1990

A 3 gigabit/second bit interleaving time division multiplexer and a time division demultiplexer for high speed computing and communications systems

Donald J. Wingate
Lehigh University

A 3 GIGABIT/SECOND
BIT INTERLEAVING TIME DIVISION MULTIPLEXER AND
A TIME DIVISION DEMULTIPLEXER
FOR HIGH SPEED COMPUTING AND COMMUNICATIONS SYSTEMS

by

Donald J. Wingate

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Electrical Engineering

Lehigh University

1990

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 9, 1990

(date)

Frank W. Hielscher

Professor in Charge

Lawrence J. Vaneni

Chairman of Department

Acknowledgments

The author would like to express his appreciation to the following people for their contributions to this thesis:

Dr. F. H. Hielscher, for his guidance and helpful suggestions.

R. W. Swartz, for his guidance and insight that proved crucial to the circuits successful frequency response over 3Gbit/s data rates.

My AT&T Management, for its support and investment that has allowed me to fabricate this design in silicon to verify its performance.

My wife, Candy, for her patience and support through the many long days and nights spent throughout my educational experience at Lehigh University.

My daughter, Brandi, for giving me that last ounce of incentive that I needed to get this project off the ground.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The never ending struggle to transmit and receive data at ever increasing data rates has pushed computing and communications systems to strive for very high speed circuits capable of handling several Gigabit per second (Gbit/s) streams of data. Two circuits which have quickly become the work horse in todays high speed transmission market place are the Bit Interleaving Time Division Multiplexer and the Time Division Demultiplexer. The Multiplexer synchronously combines two or more channels of data into a single high speed data bit stream prior to a transmitter, and then after a receiver, the Demultiplexer reduces the serial stream back into its original component data channels. Various logic architectures have been used to implement the design of these two circuits. A vast majority of these designs are presently fabricated in the Gallium Arsenide Technology because of its high electron mobility compound semiconductor wafer and low parasitic resistances and capacitances. Lately, advances in high speed bipolar silicon transistors have been focusing on reduced device parasitics by using silicon dioxide sidewalls and highly doped polycrystalline silicon for the formation of self-aligned emitter and base regions, as well as low capacitance resistors.

This paper describes the design and layout of an 8-channel-to-1 channel Bit Interleaving Time Division Multiplexer and a 1-channel-to-8 channel Time Division Demultiplexer. These circuits are targeted for the recently developed "Bipolar Enhanced Self-aligned Technology" (BEST-1) which effectively minimizes device parasitics by producing a nonoverlapping self-aligned transistor with a cutoff-frequency of 12GHz for the minimum npn transistor. Computer simulations using ADVICE (a SPICE-like circuit simulation tool) demonstrates typical operation above 3 Gbit/s for both circuits. Both the Multiplexer and Demultiplexer have compatible input and output Emitter Coupled Logic (ECL) levels and make use of an internal 400 millivolt differential ECL logic swing. These LSI circuits are slight modifications of a ripple counter controlled architecture presented by M. Ida, et al. for Nippon Telephone and Telegraph in August of 1989.[1] The two circuits which are laid out on a single die using a Cell based semi-custom approach are packaged in a 68 lead high speed multilayer ceramic TriQuint package with 50Ω controlled impedance transmission lines from the chip bond pads to the package leads. The typical power dissipation for the die is 1.44W with the Multiplexer using 46% of the power.

1. Introduction

Over the past few years, the world has been pushing the speeds at which data can be transmitted and received from rates of tens of Megabits/second (Mbits/s) to several Gigabits/second (Gbits/s). Almost daily, advancements in fiber optic computing and communications are surfacing which place aggressive requirements on circuit designers for the development of high speed circuits which can successfully transfer and receive data at these rates. For the past few years, the most efficient solution has been provided by the Bit Interleaving Time Division Multiplexer and the Time Division Demultiplexer. The Multiplexer combines two or more low frequency parallel data channels into a single high frequency serial data bit stream as in figure #1. In this fiber optic system application, the data stream is then

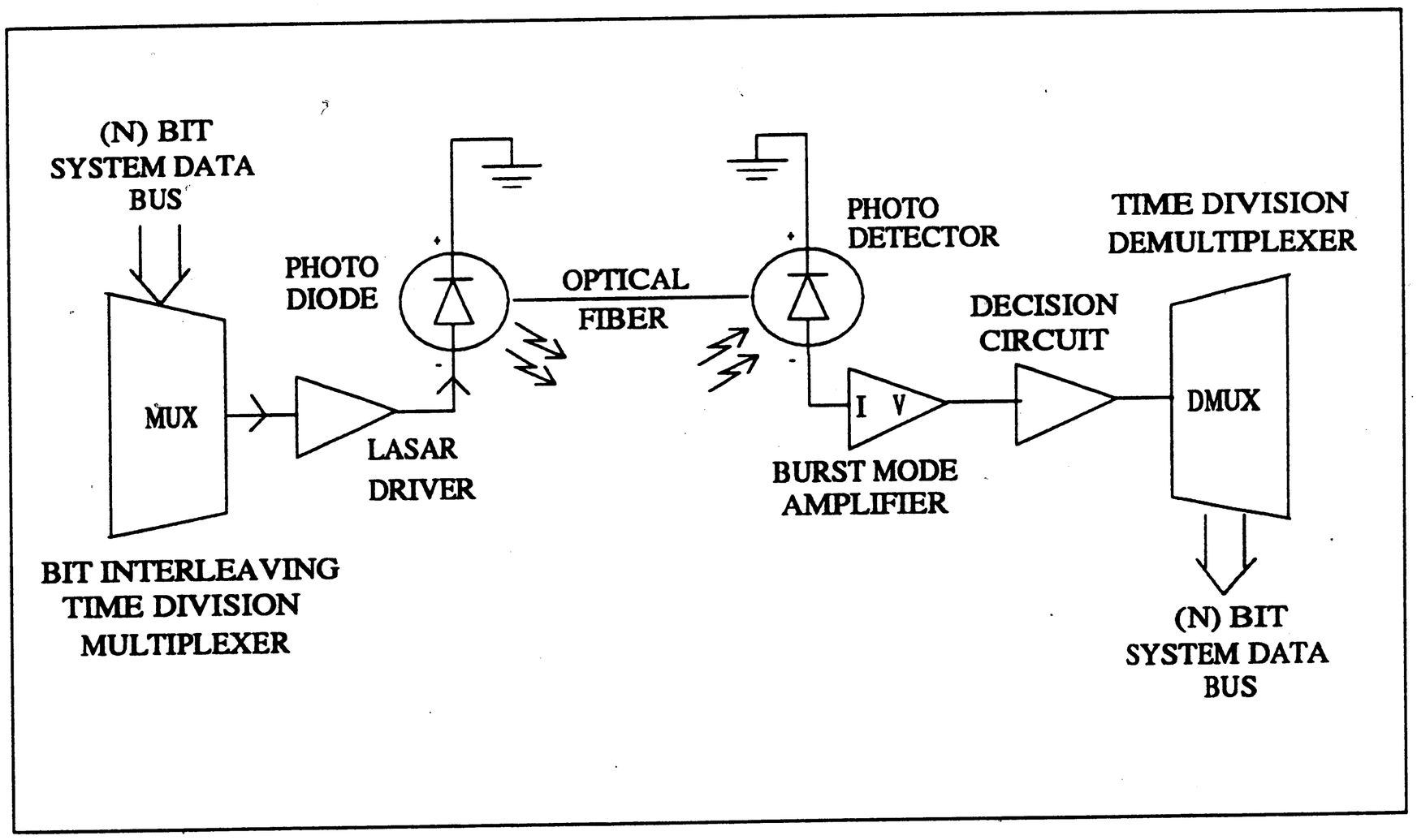


Figure #1. Fiber Optic application.

delivered to a high speed laser driver which controls a photo diode. This diode transmits the data by way of light pulses along a fiber optic cable to a second photo sensitive diode at the

receiver. This diode transforms the optical data into current pulses which are directed into a Burst Mode Amplifier. The amplifier converts the current pulses into voltage and directs the data into a Decision Circuit which amplifies the data prior to the final Time Division Demultiplexing stage. This final stage accepts the reconstructed data bit stream from the decision circuit and reduces this serial stream into the original component data channels that were input to the multiplexer prior to transmission.

An example of a Bit Interleaving data stream produced by the Time Division multiplexer is of two nibbles of shown in figure #2. Analysis of this waveform shows a continuous stream

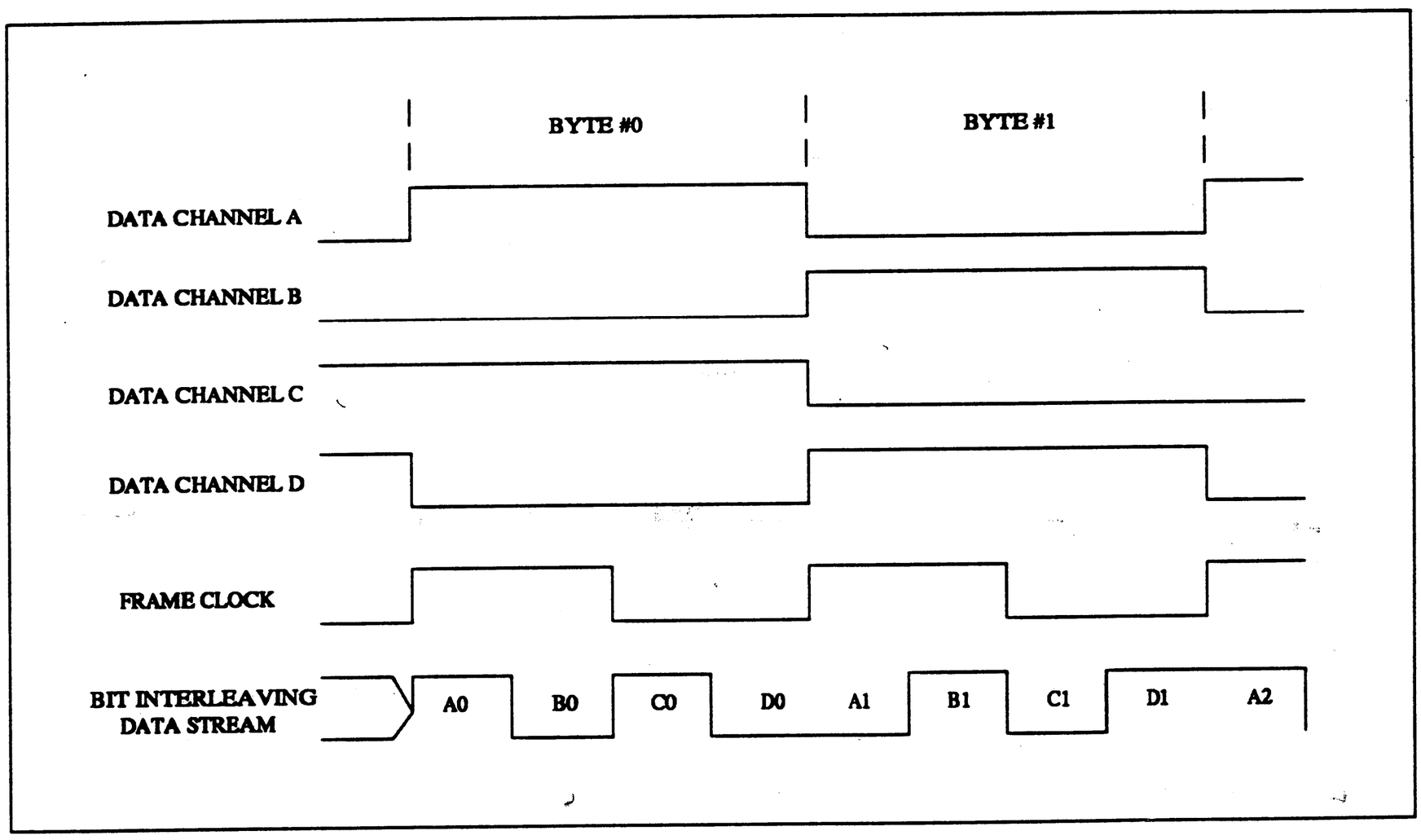


Figure #2. Bit Interleaving Data Stream

information. Note, that each nibble within the serial data stream consists of an ascending order of data bits A through D, captured in parallel from the data channels A through D during each cycle of the frame clock. Also note that the beginning of every new nibble of information is distinguished by a rising edge of the frame clock.

Since the early 1980's, the majority of these high speed data circuits have been fabricated in the Gallium Arsenide (GaAs) Technology. This technology is fabricated upon a compound semiconductor wafer, which is comprised of a Gallium and an Arsenic interpenetrating face centered cubic sub-lattice.[2] The combination of these elements results in a high electron mobility semiconductor on which can be built devices with low parasitics and fast frequency response. However, this performance is costly as a result of the processing difficulties which exist with surface oxidation, evaporation and etching.

The fabrication of integrated circuits is based on the ability of selectively introducing various dopants into a wafer at specific locations and times. Much of this is controlled by repeating the following sequence of events. First, a thick oxide is grown over the entire face of the wafer. Secondly, a photosensitive chemical (called photoresist) is applied to the surface of the wafer followed by a masking process that defines regions on the wafer that will be exposed to ultraviolet light. Thirdly, the photoresist is developed after the exposure, creating a pattern which protects certain regions of oxide while leaving others unprotected. Finally, the unprotected oxide regions are removed with a chemical etch. This exposes certain parts of the wafer surface in order that dopant atoms can be introduced to the wafer with subsequent implantations or high temperature diffusions. The key to the above sequence is the initial growth of the field oxide which serves as the basis for the remaining steps. Unfortunately, the growth of an oxide on a GaAs wafer is difficult since Gallium and Arsenic have different oxidation rates. This allows for the potential of developing a metallic phase between the interface of the GaAs wafer and the grown oxide, which electrically connects all active devices on the wafer surface. Uniform surface etching is a second processing difficulty with GaAs wafers. The Arsenic atoms along the Arsenic face of the sub-lattice have two free electrons, while the Gallium atoms along the Gallium face do not have any free electrons. Therefore, the Arsenic face of the crystal tends to be more electrically active than the Gallium face causing it to have faster etch rates and produce a smoother surface over the Gallium face. Uniform surface evaporation is a third processing difficulty which exists with GaAs wafer processing.. This occurs at temperatures below 770 degrees C when the Arsenic evaporates more rapidly from the Arsenic face than the Gallium does from the face of its sub-lattice. This creates an imbalance at the wafer surface between the number of Arsenic and Gallium atoms which further complicates the oxidation and etching processes.[2]

Silicon Technologies, on the other hand, do not have these complicated processing difficulties because the fabrication process occurs upon a single elemental semiconductor wafer. The silicon atom oxidizes very readily forming a native silicon dioxide which serves well as a protective barrier throughout processing. Furthermore, uniformity with etches and surface evaporations is easily maintained because of the single semiconductor atom instead of two as with a GaAs wafer. Although silicon's field mobility is not as high as GaAs, recent advances in photoresist with sub 1.5 micron pattern definition using Reticles and Step-and-Repeat sequences, along with reduced device parasitics and nonoverlapping structures have been realized to improve the frequency response of circuits designed in silicon.

The majority of integrated circuits fabricated on silicon wafers are designed in the Complementary Metal-Oxide-Semiconductor (CMOS) Technology which consists of an n-channel and a p-channel metal oxide semiconductor transistor. This technology is known for its ability to handle large gate count circuits as a result of the size of the minimum transistor and the low power dissipation required per gate. However, the frequency performance and ability of these transistors to drive high capacitance signal lines strongly depends upon the photolithographic process as can be seen in the analysis of the following equations which deal with the transconductance, g_m and drain current, I_D for a MOS device:[3]

$$g_m = \sqrt{2k'I_D\frac{W}{L}}$$

$$I_D = \frac{k'W}{2L}\left[2(V_{GS}-V_t)V_{DS} - V_{DS}^2\right]$$

where, $k' = \mu_n\frac{\varepsilon_{ox}}{t_{ox}}$, V_{GS} is the gate to source voltage, V_t is the threshold voltage, and V_{DS} is the drain to source voltage. Note, that in order to achieve a large transconductance for high frequency response, and a large drain current sourcing transistor for high capacitance line driving, the transistors will have to have a large width, W, and a small channel length, L, between the source and drain regions in order that the aspect ratio $\frac{W}{L}$ is large. This explains the importance of scaling the MOS transistor to define finer and finer line widths processes for the improvement of this device. To date, processes are available which have scaled the transistor channel widths to achieve reproducible 0.9μm results. In turn, the application of these devices in circuits are producing reliable designs which operate at clock frequencies up to

155 Megahertz (MHz). Many IC Houses have been struggling to improve the frequency performance of these transistors to reach circuit operations up to 300MHz. This requires finer line widths, down to the 0.5-0.6μ channel width capabilities, which rely on electron beam machines for the direct writing of the masking information onto the wafers. Although, there are presently very few reliable circuits in the market place that operate at these speeds, the rate at which processing development has progressed in the past few years suggests that devices in this speed range are not far away.

The Bipolar transistor, which is also fabricated in the Silicon Technology, produces higher frequency response and line driving capabilities over the MOS transistor. This transistor is larger and requires more power, but its transconductance and current supplying performance are orders of magnitude better than the MOS transistor as can be seen with the following equations:

$$g_m = q\frac{I_c}{kT}$$

$$I_C = I_S\, e^{\frac{qV_{BE}}{kT}}$$

where, $I_S = qAD_n\frac{n_{po}}{W_B}$ and A is the emitter area, D_n is the diffusion constant for electrons, n_{po} is the equilibrium concentration of electrons in the base, and W_B is the electrical base width. Note, that these equations demonstrate that the transconductance, g_m of the bipolar transistor depends linearly upon the collector biasing current, while the line driving collector current I_c depends inversely upon the base width of the transistor. Therefore, upon scaling the bipolar transistor using shallower junctions, the electrical base width reduces which increases the I_S for the transistor and the collector current. In turn, this improves the transconductance, and the overall cut-off frequency of the device. Therefore, the control of junction depths is very important to achieving a high performance transistor, instead of the photolithographic process which limits the MOS transistor. Over the past few years, many IC houses have been successfully controlling junction depths through ion implantation. This along with advancements with silicon dioxide walling of the base region (for reduced junction capacitances) and dual contacts to the base region (for reduced extrinsic base resistance) have led to improvements in the bipolar transistor and circuit performances with clock frequencies

up to the 500 to 600 Megahertz range.[4]

Although this performance is a vast improvement over the MOS transistor, it is still not capable of handling the Gbit/s data rates required by the Bit Interleaving Time Division Multiplexer and Time Division Demultiplexer which can be supplied by the GaAs transistor. In order to achieve these speeds, further improvements are necessary with the processing of the bipolar transistor to achiever smaller electrical base widths and additional reductions with parasitic resistances and capacitances. One approach two achieve these results is to adopt a dual "Polysilicon Self-Aligned process in which both the emitter and active base regions of the bipolar transistor are formed in a single mask step, and contact to the base is made via a doped polysilicon layer which extends over the surface of the wafer."[4] A form of this was recently developed by AT&T Microelectronics in conjunction with AT&T Bell laboratories in Holmdel, New Jersey called "BEST-1" (Bipolar Enhanced Self-aligned Technology).[5] In addition to using the highly doped polycrystalline silicon for the formation of the self-aligned emitter and base regions, this polycrystalline silicon is also used in the formation of low capacitance resistors. Furthermore, this process effectively achieves a very small base width by incorporating an oxide spacer which confines the emitter base junction and subsequent transistor action vertically under the emitter. Furthermore, this oxide spacer helps to create the nonoverlapping structure which results in reduced device parasitics. The Technology makes use of 1.5μm design rules to produce the minimum transistor which has a typical cut-off frequency, f_T, of 12GHz.

Although the work in this paper will be targeted for this process, an even better technology with high drive capability and low power dissipation is available in silicon by incorporating both the MOS and Bipolar transistors onto the same circuit. This Technology is known as BICMOS and is presently scheduled for "BEST-2". This takes the best of both worlds by using the low power MOS transistor to perform the internal logic, and using the bipolar transistor to effectively drive the high capacitance signal lines. However, since this process is not available at this time, the design of the Time Division Multiplexer and Demultiplexer will targeted for "BEST-1."

The onset of "BEST-1" comes at a time when many high speed system designers are beginning to question the short term future of Gallium Arsenide semiconductor houses. The increased

difficulties in the manufacturability of GaAs circuits have many Semiconductor houses beginning to question the capital development expenditures, especially with comparable silicon processes like BEST-1 which can handle the Gallium Arsenide circuit speeds.) Therefore, system designers are beginning to search for second source designs of Time Division Multiplexer and Demultiplexer circuits fabricated in viable processes. Specifically, the designers are interested in 2.5Gbit/s operation since many of the present day fiber optic communication systems operate at 2.5GHz internal clock speeds. The system channel capability at present varies between 8-to-1 channel and 16-to-1 channel Multiplexer and Demultiplexer applications. (This is a direct result of the present frequency capability of MOS circuits.) Those system designers who are counting on the future availability of reliable 300MHz MOS logic circuits, require the capability of combining and separating eight 300MHz data channels to achieve the 2.5Gbit/s data bit rates. On the other hand, the conservative system designers choose to rely on the present day 155MHz MOS logic circuits, and have a need for combining and separating sixteen 155MHz data channels to achieve the 2.5Gbit/s data bit streams. Independent of the channel requirements, there are presently three major Bit Interleaving Time Division Multiplexers and two major Time Division Demultiplexer architectures in the market place.

The first Multiplexer and Demultiplexer architectures are similarly controlled by a shift register scheme. "N" data channels are multiplexed by the Time Division Multiplexer by an "N" bit shift register controlled by a high speed clock and a divide by "N" version of the high speed clock. With the first rising edge of the divide by "N" clock, the data from the "N" data channels is parallel loaded into the shift register. Then, with every subsequent high speed clock edge the data is serially shifted through the register to an output buffer where it is finally transmitted to the Demultiplexer. The Demultiplexer operates opposite to the multiplexer by serially loading the incoming data into an "N" bit shift register. Once "N" bits of data are loaded into the register, the divide by "N" counter raises a data ready flag, and the data is parallel unloaded from the register onto data channels, recreating the original component data channels.

The second Multiplexer architecture involves the retiming of the multiplexed data prior to transmission. This is done with a Johnson Counter, Data latch, Shift register, asynchronous channel combining multiplexer and a delay line. First, the Johnson counter divides the

frequency of the incoming clock by the "N" number of channel inputs that are going to be multiplexed. This clock frequency is then used to drive an "N" bit data latch which captures the incoming data in parallel. Then, a second clock signal is generated from the counter by logically "ANDing" the first and second stages of the Johnson counter. This results in a narrow pulse which is input to an "N" bit shift register, which shifts the pulse with every transition of the input high speed clock. The outputs of the shift register control the select leads to the asynchronous channel combining multiplexer, such that this narrow pulse individually selects a bit of data from the "N" bit data latch. Once this data is selected, it is directed to a final retiming master slave flip-flop which retimes the data producing an output data stream with minimal edge jitter. The delay line for the high speed clock is necessary prior to the final retiming flip-flop in order that the same high speed clock edge can both propagate through the Johnson Counter, Shift register, and asynchronous channel multiplexer to select a bit of data, and then also clock that same data into the final retiming flip-flop.

The last Multiplexer and Demultiplexer architectures involve a cascade of master slave flip-flops which have their select and clock leads controlled by a ripple counter. A good example of this was recently published in August of 1989 by Nippon Telephone and Telegraph (NTT). The 16 to 1 channel Multiplexer consists of a cascade of 2-to-1 Multiplexing master slave flip-flops controlled by a 4 bit ripple counter which is clocked by the high speed input clock. The first stage has 8 of the master slave mux flip-flops which have their data select inputs controlled by the divide by 16 output from the 4th stage of this ripple counter. The 3rd stage of the ripple counter outputs a divide by 8 output which clocks these 8 flip-flops capturing each bit of data from the two data channels that are input to the flip-flop. Therefore, this first stage manages to merge the 16 data channels to the 8 outputs of the flip-flops. The second stage, consisting of 4 master slave mux flip-flops, have their select leads controlled by the divide by 8 output from the ripple counter while the divide by 4 output performs the clocking. Since the divide by 8 output clocks the first stage and selects data for the second stage, then the original 16 data channels which were merged onto the 8 channels gets merged again to the 4 data outputs from the second stage. This sequence is repeated in the 3rd stage which consists of 2 master slave mux flip-flops that combine the 16 data channels that have been merged to 4 channels, to 2 channels. The final stage of the Multiplexer consists of a single master slave mux flip-flop, which individually selects a bit of the 16 data channels from the 2 channels,

creating a final Bit Interleaved serial output data stream with minimal edge jitter.

The Demultiplexer operates in reverse of the Multiplexer. The 4 bit ripple counter is again controlled by the input high speed clock, and the divide by 2 output is used to clock a parallel set of master slave flip-flops which represent the first stage. One of the master slave flip-flops is preceded by a positive level latch making it in effect a three stage master slave flip-flop which captures data on the falling edge of the divide by 2 clock, but delays the output of that data from the latch until the subsequent rising edge of the clock. Therefore, as the Bit Interleaving data stream is input in parallel to both of these flip-flops, on falling edges of the divide by 2 clock a data bit will be captured by the three stage flip-flop. Then on the subsequent rising edge this data bit will be transferred to the second stage of the Demultiplexer while the other parallel master slave flip-flop captures the next input data bit and transfers it to the next stage without delay. The second stage consists of two separate first stages which are clocked in parallel by the divide by 4 output from the ripple counter. One of the stages receives the data output from the three stage flip-flop while the other stage receives data from the master slave flip-flop. Therefore, with clock edges from the divide by 4 clock, this stage separates the 2 data channels to 4, which it directs to the third stage of the Demultiplexer consisting of 4 separate first stages clocked in parallel by the divide by 8 output from the ripple counter. This stage repeats the above sequence creating 8 data channels from the 4 data channels, which it then directs into the final Demultiplexing stage consisting of 8 separate first stages clocked in parallel by the divide by 16 output from the ripple counter. This final stage recreates the original 16 data channels which were input to the Multiplexer, by separating the 8 input data channels input to this stage by the previous third stage.

For the designs to be described in this paper, the final ripple counter architectures will be adopted for both the Multiplexer and Demultiplexer. The other architectures controlled by the shift register approach and retiming method will not perform as well as these architectures as a result of the high fanout loading on the input high speed clock. This is most evident with the shift register architectures where the fanout loading increases linearly with increases in channel count. Although the retiming architecture makes an attempt of reducing the fanout loading, the increased difficulty in accurately controlling the delay line through manufacturing from lot to lot eliminates this approach. Therefore, this paper describes the design and layout of an 8-to-1 channel (8:1) Bit Interleaving Time Division Multiplexer and 1-to-8 channel (1:8) Time

Division Demultiplexer which will handle typical data rates greater than 3Gbit/s. Both circuits will have compatible Emitter Coupled Logic (ECL) level inputs and outputs as well as an internal 400 millivolt (mV) ECL logic swing. The circuits will be fabricated as a single die in AT&T's silicon Bipolar process (BEST-1) discussed above. A Cell based semi-custom approach will be taken with the layout of the circuit which will sacrifice silicon area for quick layout turn around. Although this will not provide the most optimum layout for maximum frequency performance because of the potential of many unused devices, it will allow for maximum layout flexibility and security in circuit corrections in case first silicon is inoperative at chip test. The design techniques and circuit architectures described in this paper are directly expandable for higher or lower channel capability Time Division Multiplexer and Demultiplexer circuits to customize the circuit to handle various frequency performance CMOS logic other than the 300MHz logic for which these designs are targeted.

2. Specifications and System Requirements

A successful fiber optic system design which incorporates a Bit Interleaving Time Division Multiplexer and a Time Division Demultiplexer requires certain system specifications and requirements dealing with circuit speeds, input and output data specifications to and from the Time Division circuits, as well as the alignment of this to system clocks and output data channels.

First, the 2.5GHz system clock speed discussed in the introduction is quickly becoming an industry standard for fiber optic systems in the world. This clock signal is generated from quartz crystal based oscillators which produce an Emitter Coupled Logic (ECL) level sinusoidal signal with an 800 millivolt (mV) amplitude peak-to-peak (p-p). In addition to this, the laser driver and decision circuits from figure #1 are typically designed in ECL with ECL compatible inputs and outputs. Based on the system clock speed and these circuit interfaces, it makes sense to design both Time Division circuits in ECL with ECL compatible inputs and outputs. This poses a minor problem with the input data channels to the Multiplexer and the output data from the Demultiplexer which typically come from and go to CMOS logic operating at either 155MHz or 300MHz data rates as discussed in the introduction. However, it makes more sense to convert these CMOS levels to ECL levels than attempt to manipulate the other ECL level inputs and outputs.[6]

Second, as a result of the finite space on system boards, the number of Time Division circuits incorporated in the system must be minimized. This not only results from the power and cooling capacity within the system, but also from the available frequency of the CMOS logic. If the logic used in the system produces 155MHz data rates, then 16 data channels can be multiplexed to achieve the 2.5GBit/s data bit stream, and then demultiplexed to recreate the original 16 channels after transmission. However, if the CMOS logic used in the system produces 300MHz data rates, then only 8 data channels need to be multiplexed and demultiplexed. Therefore, a system with faster CMOS logic will require twice as many Time Division circuits than a system with the slower 155MHz logic. Independent of which channel count is chosen, the alignment of the data to be multiplexed and the data to be demultiplexed to the high speed clock is very important. Here again, available board space dictates that only one delay line can be used for the alignment of these data with the high speed clocks.

Actually, this brings up another reason why the CMOS data logic levels should be converted to ECL levels to ease the data alignment between these signals and the ECL level high speed clock inputs.

Finally, output data from the Time Division circuits is also important. With the Multiplexer, it is important that the output data bit stream have minimal edge jitter so that the laser driver can accurately control the photo diode. Any edge jitter caused by variations in the zero point crossings of the output data signal from the multiplexer will lead to inaccuracies in the transmitted light pulses. Therefore, the width between single logic high and logic low pulses must be controlled as well as minimizing the variation in width of a double wide pulse to two single wide pulses. In addition to this, peak-to-peak jitter is equally as important and must be minimized as well. With the Demultiplexer, edge jitter is not as important as skew. Minimal skew is required between the output data from all of the channels in order that a single delay line can be used with the receiving CMOS logic to capture all of the data in parallel. Furthermore, if the data being output from the Demultiplexer is coming out during the wrong data byte or on the wrong data channel, then the Demultiplexer must have a provision to realign this data in time and space to the correct data channel during the correct data byte.[6]

Based upon the above requirements for a successful system design, the following circuit specifications and requirements will be met by the Time Division Multiplexer and Demultiplexer designs of this paper.

A. Multiplexer and Demultiplexer

1. Both circuits must operate at input clock speeds greater than 2.5GHz over 10% power supply variations, temperature variations from 0 to 100 Degrees C and worst case processing variations in transistor gains, sheet resistances and junction capacitances.

2. Typical power dissipation of less than 1.5 Watts for the single die (including both circuits) will satisfy system thermal requirements.

3. Both circuits will operate from an externally produced ECL level differential sinusoidal clock signal 800 millivolts (mV) peak to peak (p-p).

4. Not more than one external delay line will be permitted for each circuit for the alignment of the clock to the input data.

B. Multiplexer Specifics

1. Input data channels are 300MHz ECL level single ended signals 800mV p-p and they must all be latched into the circuit in a parallel fashion to minimize the number of delay lines.

2. Output data is an ECL level differential signal with 800mV +/- 200mV p-p into a 50Ω load. Rise and fall times should be less than 150picoseconds (psec) from 10% to 90% of the signal swing. The data crossover point should lie between 30% to 70% of the amplitude. Under typical conditions, there should be less than 10% edge jitter resulting from deviations in zero crossing intervals.

C. Demultiplexer Specifics

1. Input data is a ECL level differential sinusoidal signal with 800mV p-p.

2. Output data is latched and arrives in parallel at all eight outputs with less than 100psec skew between signals, and less than 2 nanoseconds (nsec) skew between any data output to the rising edge of the frame clock output.

3. The circuit must include a provision for data framing. (i.e- the data output from the Demultiplexer can be shifted in time and space for alignment with the proper output channel and time slot.

The next chapter will more closely analyze the circuit architectures briefly discussed in the introduction and will compare each design with the above specifications and requirements to see how each would operate in a system.

3. Circuit Architectures

A literature review of many Bit Interleaving Time Division Multiplexers and Time Division Demultiplexers as discussed in the Introduction shows three major types of Multiplexer architectures and two major Demultiplexer architectures.

The first type of Multiplexer and Demultiplexer circuits shown in figure #3 are very similar in architecture. Both circuits utilize a shift register and divide by "n" counter to control the input

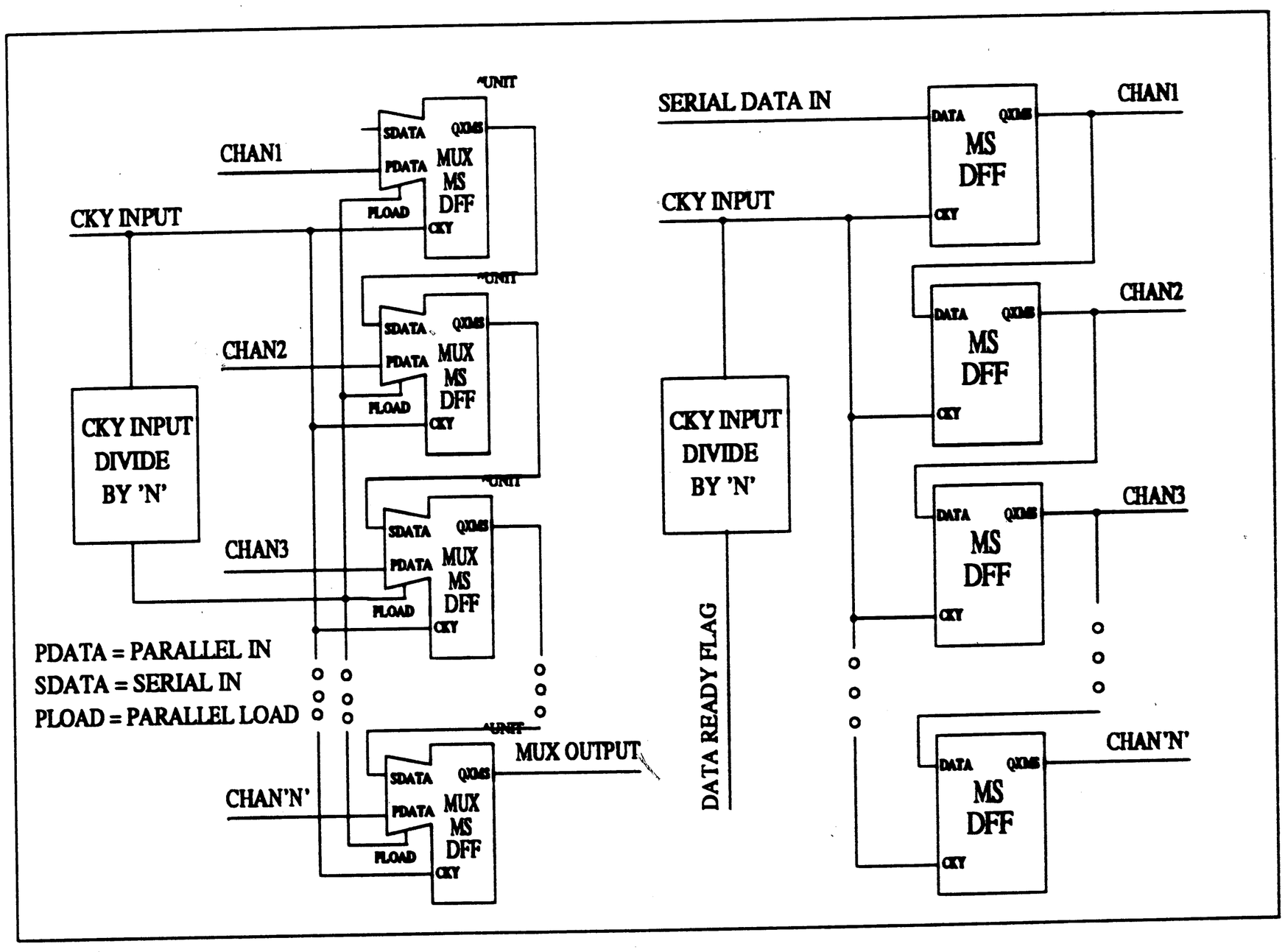


Figure #3. Multiplexer and Demultiplexer Shift Register Architecture

and output of data.[6] In the multiplexer, "n" channels of data are parallel loaded into an "n" bit shift register under the control of the divide by "n" counter. After the data is loaded, the input

clock transmits the bits of data by serially shifting the data through the shift register. The demultiplexer operates in reverse by serially loading the shift register with the input data. When the register is full, the divide-by-n counter raises a data ready flag and all "n" bits of data are parallel output from the shift register. Analysis of this architecture shows that all of the specifications and system requirements previously listed are covered, except for the provision for output data alignment with the demultiplexer. It is feasible though to modify the divide-by-n counter to hold counts and delay the data ready flag which is a satisfactory solution to the requirement. However, the additional logic required to perform this function, as well as the expansion of this architecture to handle larger data channel capabilities, would only worsen the speed limiting condition caused by the high fanout condition on the input high speed clock. Therefore, this architecture will not be used for the designs in this paper.

The second type of multiplexer architecture involves the retiming of data with a fixed delay line prior to the data being transmitted from the circuit as shown in figure #4. Although this architecture is expandable and can be used to implement "n" channel circuits, only the above 4 channel Multiplexer reported by G. Flower will be examined.[7] In the Multiplexer, the input clock drives a 2 bit Johnson counter which provides two divide by 4 clock frequencies which are 90 degrees out of phase. The second stage of the counter is then used to clock a "4" bit register which parallel loads the data from 4 input data channels. Meanwhile, the two clock phases are logically AND-ed creating a narrow pulse which is directed to the data input of a 4 bit shift register which controls the select inputs of an asynchronous 4 to 1 multiplexer. As the pulse is propagated through the shift register, it individually selects each of the 4 outputs of the 4 bit register that is holding the input data byte. After the last bit of data is selected, a new pulse from the AND gate enters the shift register to begin the process, similar to a single bit barrel shifter. Each of the data bits which are selected propagate through the asynchronous multiplexer and are input to a final D-type master slave flip-flop controlled by a delayed version of the high speed clock.

A closer analysis of the critical timing shows that the same input high speed clock edge which controls the selecting of the data bit by way of the shift register will also transmit that same

data bit by clocking the final retiming flip-flop. In order for this to occur, the delay line in

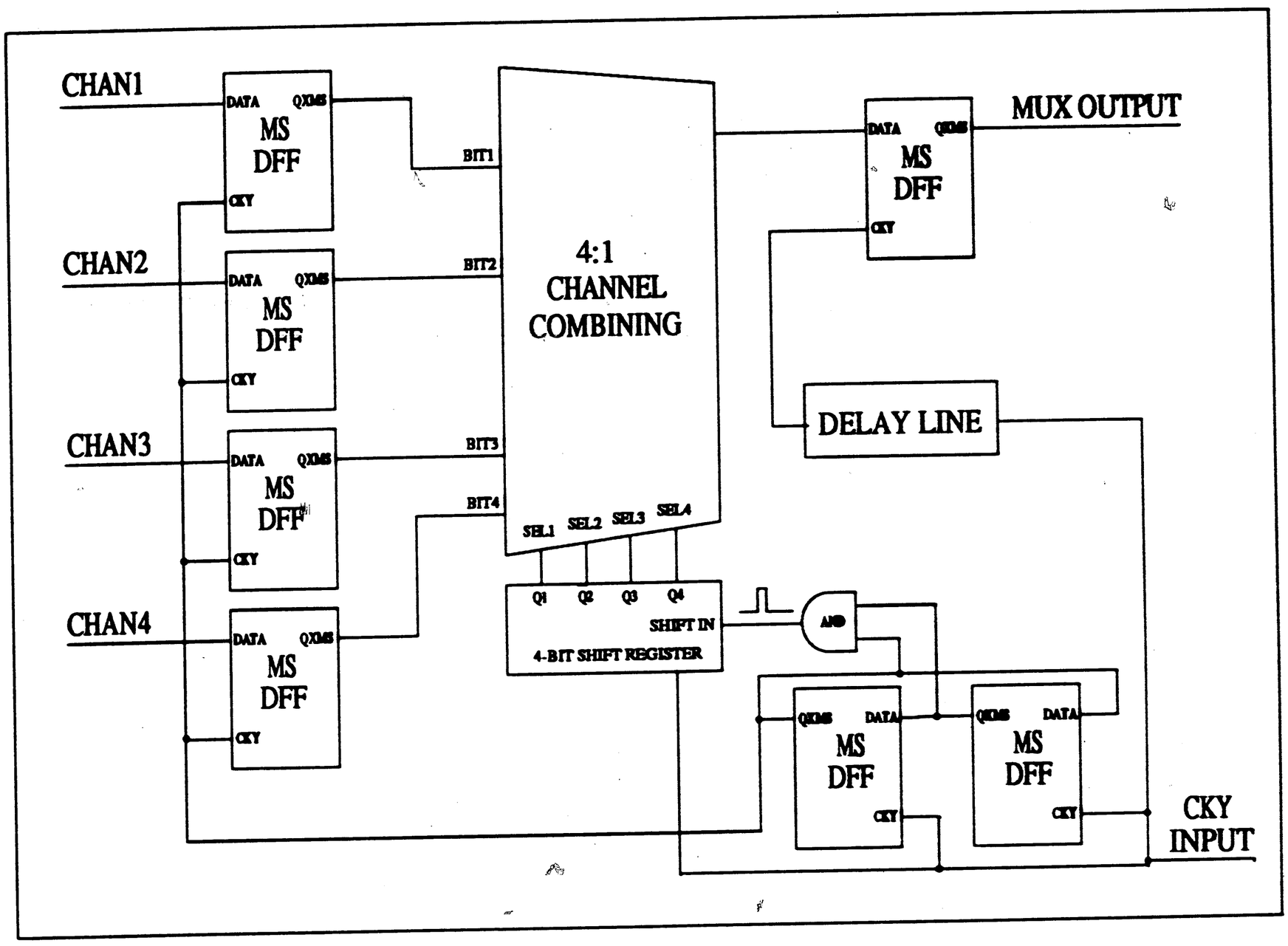


Figure #4. Multiplexer Retiming Architecture

the above circuit must be equivalent to the delay the input clock experiences propagating through the the Johnson counter, one bit of the shift register, the asynchronous multiplexer, and the data setup time to the retiming flip-flop. All of this must occur before the selected bit arrives at the final flip-flop to be retimed by a delayed form of the same clock edge which selected the data. This architecture, improves upon the speed limitation due to high fanout of the input high speed clock over the previous architecture. But, to do this, it incorporates a very critical timing relationship which G. Flower questions as to its control through manufacture from lot to lot. Furthermore, good engineering practices would suggest that the

architecture of both Time Division circuits be similar for tracking purposes. Therefore, this architecture will not be used for the Multiplexer approach.

The third Multiplexer and Demultiplexer Architecture to be reviewed was recently published in

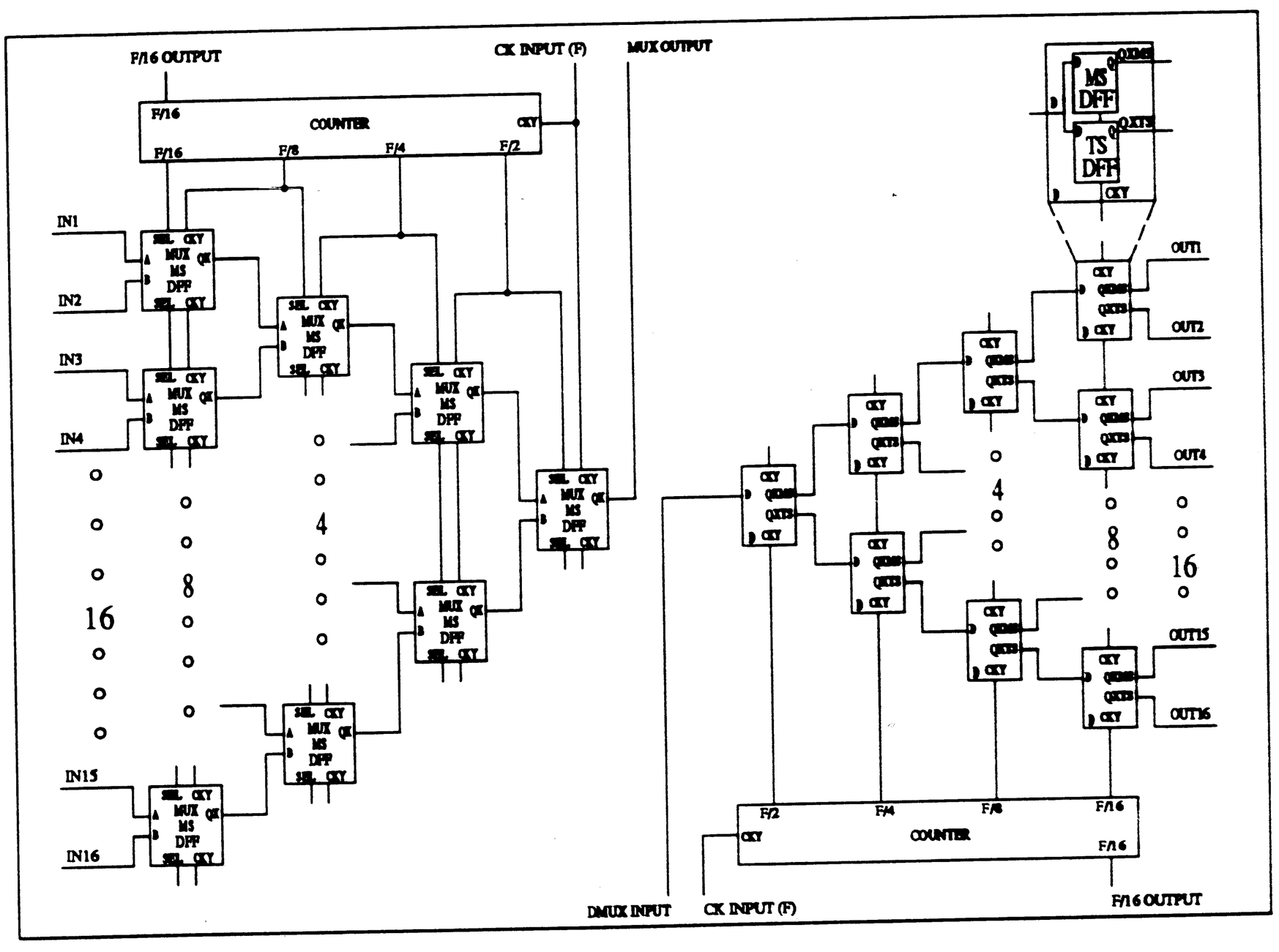


Figure #5. Multiplexer and Demultiplexer Ripple Counter Architecture

the August 1989 issue of the IEEE Journal for Solid-State Circuits.[1] These designs encorporate a ripple counter to control the selecting and clocking of a cascade of data latches as shown in figure #5. The multiplexer uses a tree of mux flip-flops which are clocked at twice the frequency as the select rate. Therefore, the first stage of mux flip-flops combines 16 channels of data into 8 by selecting each mux flip-flop at 1/16th the input clock frequency, and clocking

all of the mux flip-flops at 1/8th the clock frequency. This is repeated 3 more times with the 1/4th, half, and the actual input frequency clocks, combing the 8 channels to 4, the 4 channels to 2, and finally the 2 channels to 1 which is then transmitted from the circuit.

The Demultiplexer is similar in operation to the multiplexer except it uses D-type master slave flip-flops and a novel tri-stage flip-flop (a positive level enable latch in series with a D flip-flop as shown in figure #19 of the next chapter). In this circuit, the input data is directed to both the D flip-flop and the tri-stage flip-flop in parallel. The input clock is divided by 2 and fed to both of these gates. On the falling edges of the divide by 2 clock, the tri-stage flip-flop will latch and hold the input data, while the D flip-flop will begin to acquire new input data. With the next rising edge of the clock, the D flip-flop will latch and transmit whatever input data is located at its input, while the tri-stage flip-flop will latch and transmit the data that it had latched earlier. Therefore, the first stage of the Demultiplexer separates the input data into two data bit streams each of which is directed into another stage of parallel D and Tri-stage flip-flops clocked by a divide by 4 version of the input high speed clock. This repeats the same function, done by the previous stage, at half the frequency, therefore creating 4 data streams from the 2. These 4 data streams are then directed to a third stage of parallel D and Tri-stage flip-flops clocked by a divide by 8 version of the input high speed clock. This separates the 4 data streams to 8 which then go through the last stage of flip-flops clocked by a divide by 16 version of the input high speed clock. This final register transmits the now 16 output data channels from the circuit.

Although this architecture does not have a provision for the alignment of the output data with the Demultiplexer, it does have the benefit of having a low fanout of the high speed clock which yields faster circuit performance over the first architecture. Furthermore, the distribution of the various clock frequencies, which are derived from this clock, work out such that the faster frequencies fanout to fewer places. This architecture also does not have the difficult timing delay situation which exists with the second architecture, since all of the control is with a simple ripple counter. Therefore, the only major drawbacks to this architecture are the lack of a provision for the shifting of data in time and space for the Demultiplexer circuit and the high power dissipation which results from the large usage of data latches in the design.

In summary, of the various architectures discussed above, the ripple counter architecture presented by M. Ida, et al. is clearly superior. Therefore, this architecture will be used in the design of the 8:1 Bit Interleaving Time Division Multiplexer and 1:8 Time Division Demultiplexer. However, provisions will be included in this work to satisfy the system requirement for data alignment with the Demultiplexer circuit. Furthermore, a power reduction will be realized over a comparable 8:1/1:8 version of the existing M. Ida, et al. designs in that 18 fewer latches will be used across both circuits by carefully replacing flip-flops with single stage data latches. This produces a 5% reduction of power in the Multiplexer and an 8% reduction of power in the demultiplexer.

4. Background for Circuit Building Blocks

In order to achieve the required 2.5Gbit/s data bit rates, the 8:1 and 1:8 Time division Multiplexer and Demultiplexer circuits will be designed with an internal 400mV differential signal swing Emitter Coupled Logic approach. This is is the fastest logic form for bipolar technologies, since other logic forms like Resistor Transistor Logic (RTL), Diode Transistor Logic (DTL) and Transistor Transistor Logic (TTL) drive their transistors into saturation, resulting in increased propagation delay times. On the other hand, Emitter Coupled Logic is a nonsaturating logic. The transistors are operated between the linear and cut-off regions and are used to direct current flow within the circuit. For example, let's analyze the following differential amplifier, shown in figure #6. The circuit consists of two identical transistors, commonly referred to as an emitter coupled pair, two 2kΩ load resistors, and a regulated 400microamp (μA) current source, known as the tree current for the circuit. This circuit is powered by a single -5.2 volt power supply connected between ground and VEE (i.e- VEE = -5.2 volts). The input to the circuit drives the base of transistor Q1, while the base of transistor Q2 is connected to a regulated reference voltage of -1.2 volts. To begin the analysis of this circuit, assume that the input signal is an 800mV p-p voltage waveform centered at -1.2 volts. When the input signal is -0.8 volts, Q1 is biased into its active region of operation with its V_{BE} equal to 0.8 volts. The resulting voltage on the

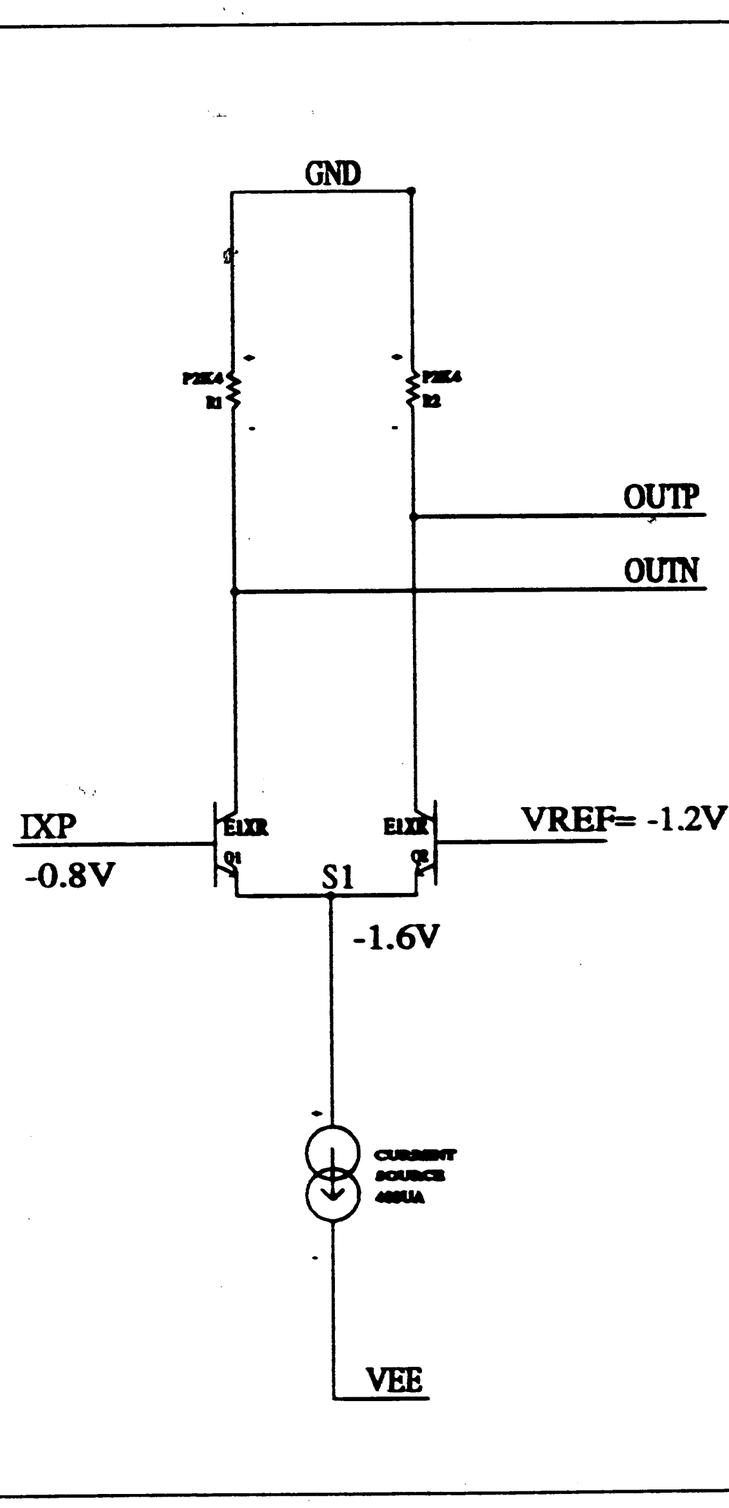


Figure #6. Single-Sided Amplifier

summing node S1 is -1.6 volts since it is a V_{BE} voltage drop below the input by way of transistor Q1. Since, the base of Q2 is fixed at -1.2 volts, the corresponding V_{BE} across transistor Q2 is 0.4 volts which biases Q2 into the cut-off region of operation. (A V_{BE} of 0.7 volts is required for a transistor to enter cut-in and 0.8 volts to become fully biased into the active region of operation.) With Q2 cut-off, all of the tree current is directed through the emitter and collector of transistor Q1 assuming infinite gain. This 400μA collector current then passes through the 2kΩ load resistor R1 resulting in an output voltage of -0.8 volts on OUTN referenced to ground. The voltage level on the other output node OUTP is equal to ground since transistor Q2 is in cut-off and no current flows through the load resistor R2.

We will now analyze the circuit as the input voltage descends to a logic low of -1.6 volts. First, when the input voltage reaches -1.2 volts the bases of the emitter coupled pairs are equivalent. At this point, the voltage on the summing node S1 is -2.0 volts and both transistors Q1 and Q2 become biased into their active regions. Since the emitter areas are equal, their V_{BE} voltages will be equal and the tree current will be equally shared between both transistors (i.e- 200μA in each transistor). Each of these currents then passes through the 2kΩ load resistors resulting in identical output voltages on nodes OUTN and OUTP of -0.4 volts. This input condition where the input voltage matches the reference voltage is called the cross over point or middle of the transition region. P. Gray and R. Meyer explain in Figure 3.27 in "Analysis and Design of Analog Integrated Circuits" that the voltage width for this transition region which guarantees that 95% of the tree current has switched from one side of the emitter coupled pair to the other is +/- $3V_T$ or +/-75mV on either side of the reference voltage.[3] Therefore, as the input voltage continues to drop to a logic low of -1.6 volts, once it is 75mV below the reference voltage the circuit will have changed state. An analysis of this state, assuming the input voltage of -1.6 volts, shows transistor Q1 biased into its cut-off region since its V_{BE} equals 0.4 volts. Transistor Q2 becomes active and holds the summing node S1 at -2.0 volts, which is a V_{BE} drop below the reference voltage. Now all of the tree current is directed through Q2 and its load resistor R2, resulting in an output voltage of -0.8 volts on output OUTP as was seen earlier with output OUTN. Furthermore, the voltage on OUTN goes to ground since the current flow through resistor R1 is stopped by transistor Q1. In summary, the 800mV input signal swing described above results in the output of two complementary 800mV signal swings from the circuit. The signal output from node OUTN is an inverted

sense of the input signal while the signal output from node OUTP is a noninverted sense. This circuit is considerably faster than any DTL, RTL, or TTL circuit because of its small signal swing, nonsaturating logic, and the small transition region of only 150mV that the input needs to traverse in order to switch logic states. This circuit is an amplifier with a gain of 5, and an output impedance of slightly less than the 2kΩ load resistors. Although DTL, RTL and TTL circuits do not have the benefit of this small signal swing, they do have compatible input and output logic levels such that their circuits output can drive their own input or a similar input of another gate. The above differential amplifier's output levels are shifted 0.8 volts higher than its input logic levels which requires modifying this circuit for digital logic.

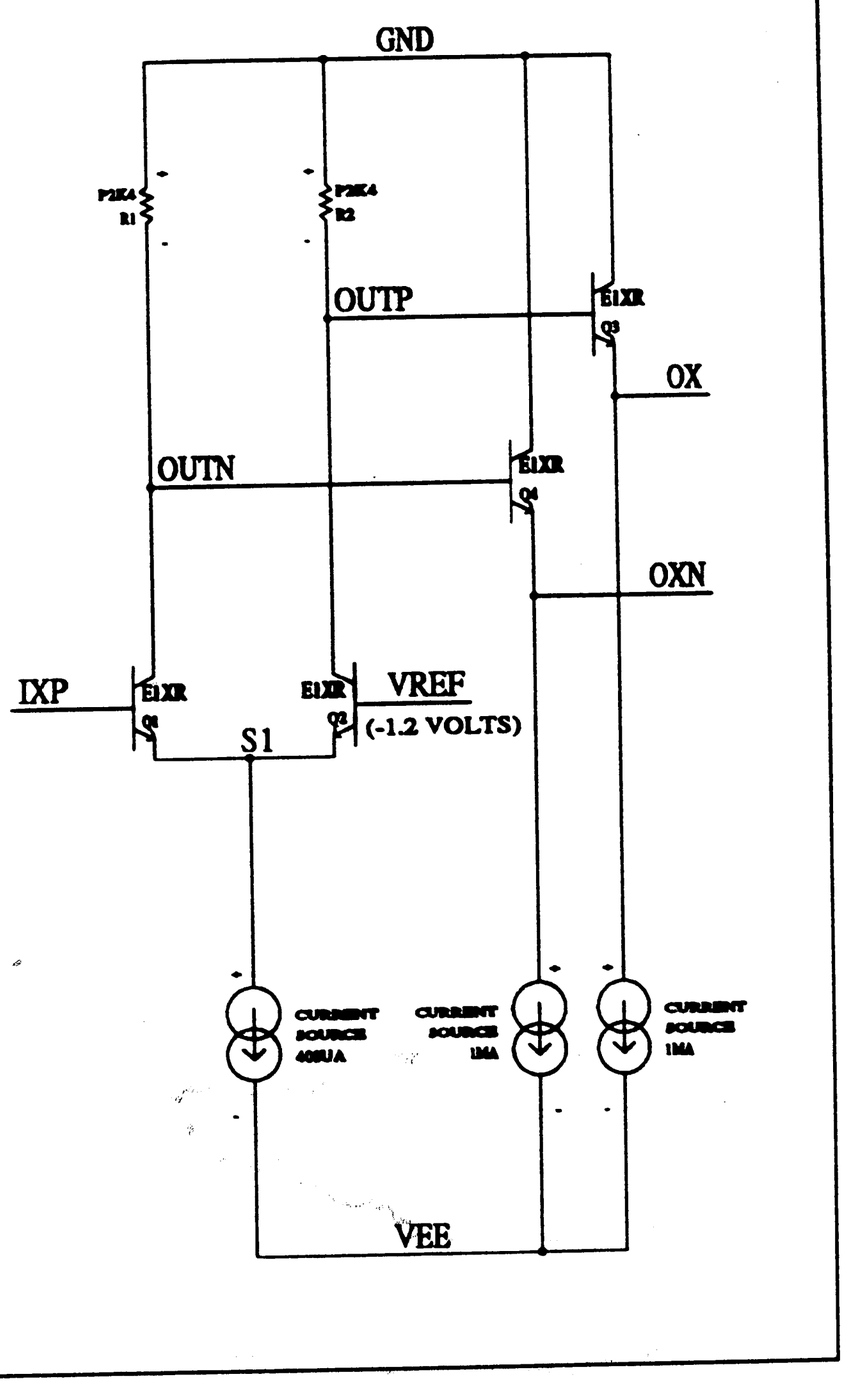


Figure #7. Single-Sided ECL Inverter

This incompatibility can be corrected and a lower output impedance can be obtained by adding a common collector (emitter follower) amplifier following the OUTP and OUTN outputs to obtain the single sided ECL inverter shown in figure #7. Analyzing this circuit shows, as expected, that the outputs OX and OXN follow the input voltages OUTP and OUTN, and produce similar 800mV signal swings, which however are shifted down the V_{BE} of transistors Q3 and Q4. For example, when the output OUTN is at 0.0 volts, the output OXN at the emitter of the emitter follower transistor Q3 is at -0.8 volts, and when the output OUTN equals -0.8 volts for a logic low, the current source pull down on OXN helps discharge the node and pull the voltage down to -1.6 volts. In

addition to adjusting the output logic levels to make them compatible with the input logic levels, the emitter follower transistor does provide two other benefits. First, the emitter follower transistor improves the output rise time for the gate by providing a lower output impedance over the previous amplifier circuit. This improves the rise time constant for the output node which is RC limited. Furthermore, the emitter follower provides an active output current gain stage, which allows for increased fanout over the load resistor of the previous amplifier.

The circuits described above are known as Single-Sided circuits because the input waveform is compared to a fixed voltage reference on the emitter coupled pair. The speed performance of these circuits are limited to the RC charging which occurs at the load resistor. One way

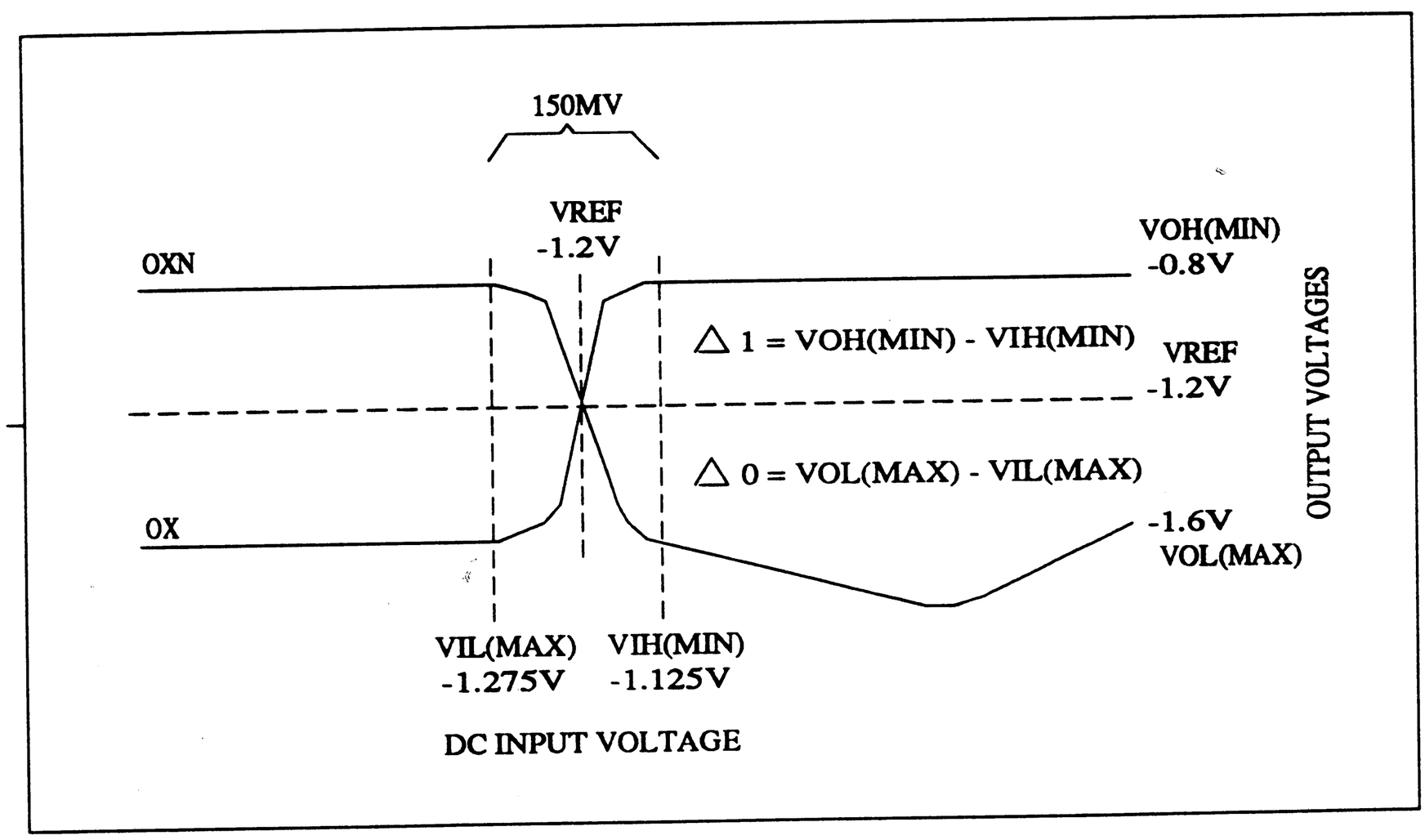


Figure #8. Typical ECL transfer characteristics.

of improving the performance is to reduce the load resistor in half, and improve the RC charging time on this node by two. However, this also effects the noise margins for the gate

by reducing the output signal swing from 800mV to 400mV. Figure #8 shows the effective noise margins for the single sided ECL inverter of figure #6. For a logic high and low the margins are identical and equal to 325mV. Unfortunately, upon reducing the signal swing to 400mV and recentering the reference voltage to -1.0v, the noise margins drop to 125mV leaving very little room for power supply noise immunity. A solution to this reduced noise margin is to replace the fixed reference source with an input signal which is the complement of the input signal IXP as shown in figure #9. Now, this differential ECL inverter has a Voh(min) = -0.8v, Vol(max) = -1.2v, Vih(min) = -1.125v, and Vil(max) = -0.875v, resulting in equivalent noise margins of 325mV again for a logic high and low, while taking advantage of the reduced signal swing and improved speed performance. Note, that the operation of this gate is the same as the previous single-sided ECL inverter, except now, the voltages on both bases of the emitter coupled pairs will be moving to direct the flow of the tree current through the circuit.

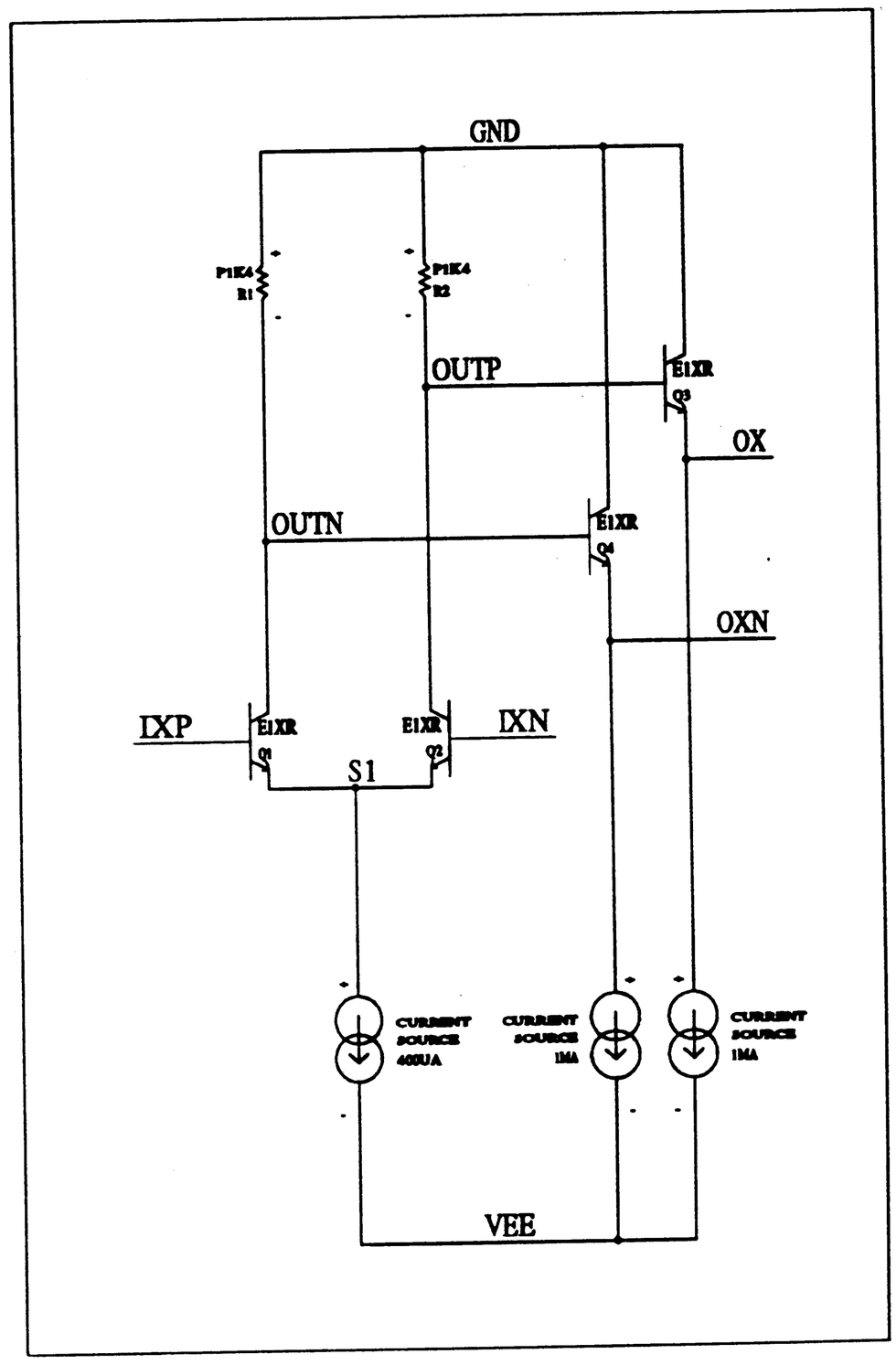


Figure #9. Differential ECL Inverter

This Differential ECL Inverter is the simplest of the ECL building blocks which can be realized. The AND/NAND function, which is a simple analog multiplier, can also be implemented by using a stacked building technique with the differential ECL inverter as shown in figure #10. Analysis of this gate demonstrates that the only input condition which will produce a logic high at the AND output OX is when both inputs IXP and IYP are logic highs.

This occurs because it is the only case where both transistors Q2 and Q6 are cut-off and stop the flow of tree current through resistor R2. If either input IXP or IYP are logic lows, the tree current is directed through resistor R2 by way of transistor Q2 or Q6 depending on whether Q5 is active or cut-off. This in turn pulls the AND output OX low and the NAND output OXN to a logic high state.

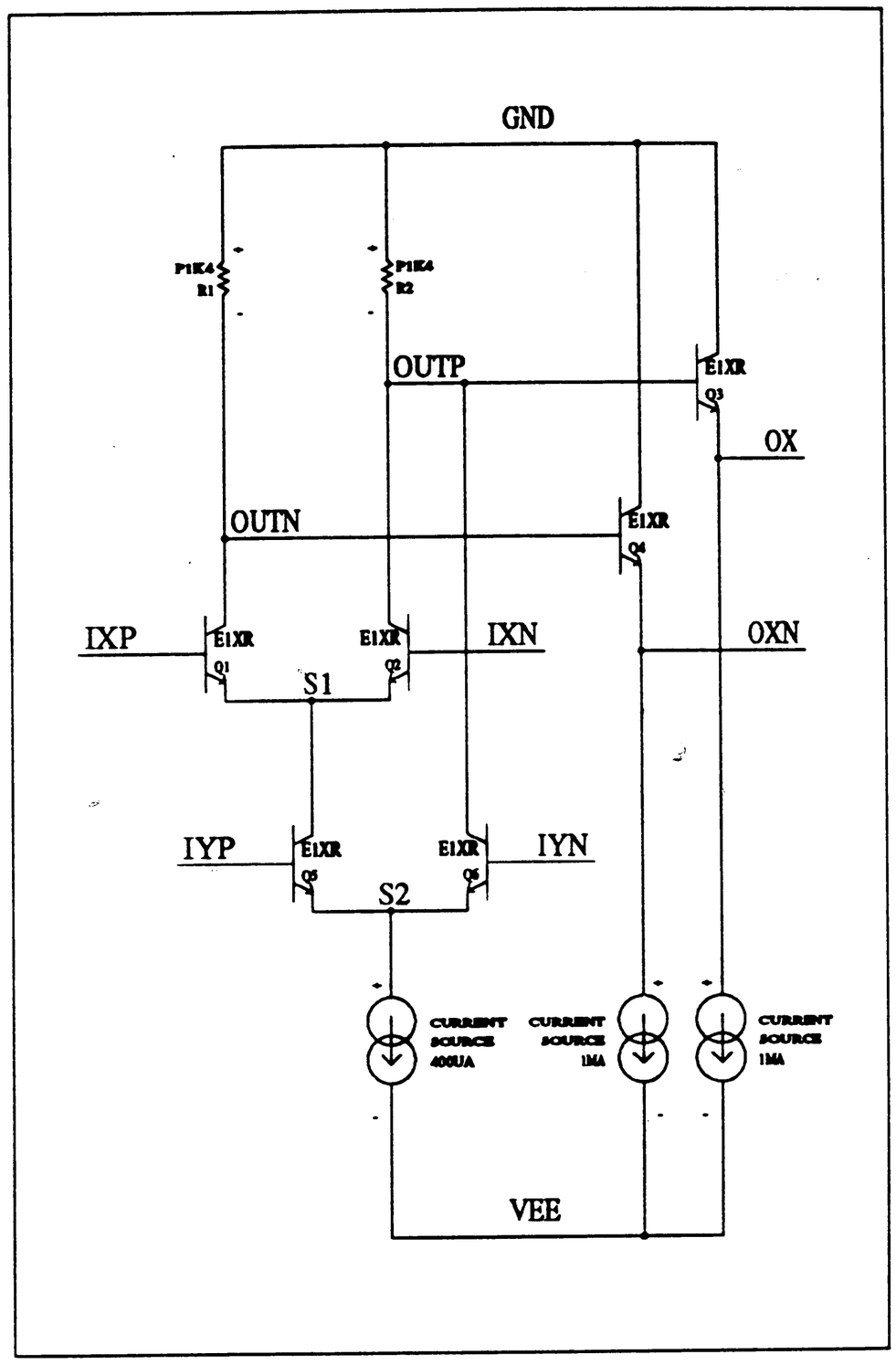


Figure #10. Differential ECL AND/NAND Gate

In the AND/NAND gate, the "IYP/IYN" input levels must be a V_{BE} drop below the "IXP/IXN" input levels. This is necessary to guarantee that transistor Q5 will not become saturated and store charge in its base region. As shown earlier, the voltage on the summing node S1 equals -1.7 volts during worst case switching. If the IYP/IYN input levels voltage swing is -1.6 to -2.0 volts, then the base-collector junction of transistor Q5 will never become more than 100mV forward biased during worst case switching which is not enough forward bias to saturate the transistor. One way to obtain this lower logic swing, is to place another common collector amplifier in front of the IYP and IYN inputs to level shift an OX type output level down one V_{BE}. A second method is to add a diode to both common collector amplifiers in the outputs of the gates as shown in figure #11.

This approach to producing the "Y" logic levels is more efficient than the common collector amplifier approach. It does not require two extra current sources because it maximizes the use of the load current source in the output stage as both a pull down and level shifter current

source. This savings in power becomes much more apparent in high fanout paths, like clock signals, where two current sources are saved with each driven gate. But, care must also be exercised with this approach for the level shifting diode adds series resistance to the OY/OYN outputs which can become substantial as the collector current increases by increasing the pull down current source. Furthermore, the diodes add parasitic junction capacitances to the OX/OXN outputs as well, which gives up some performance improvements for power savings.

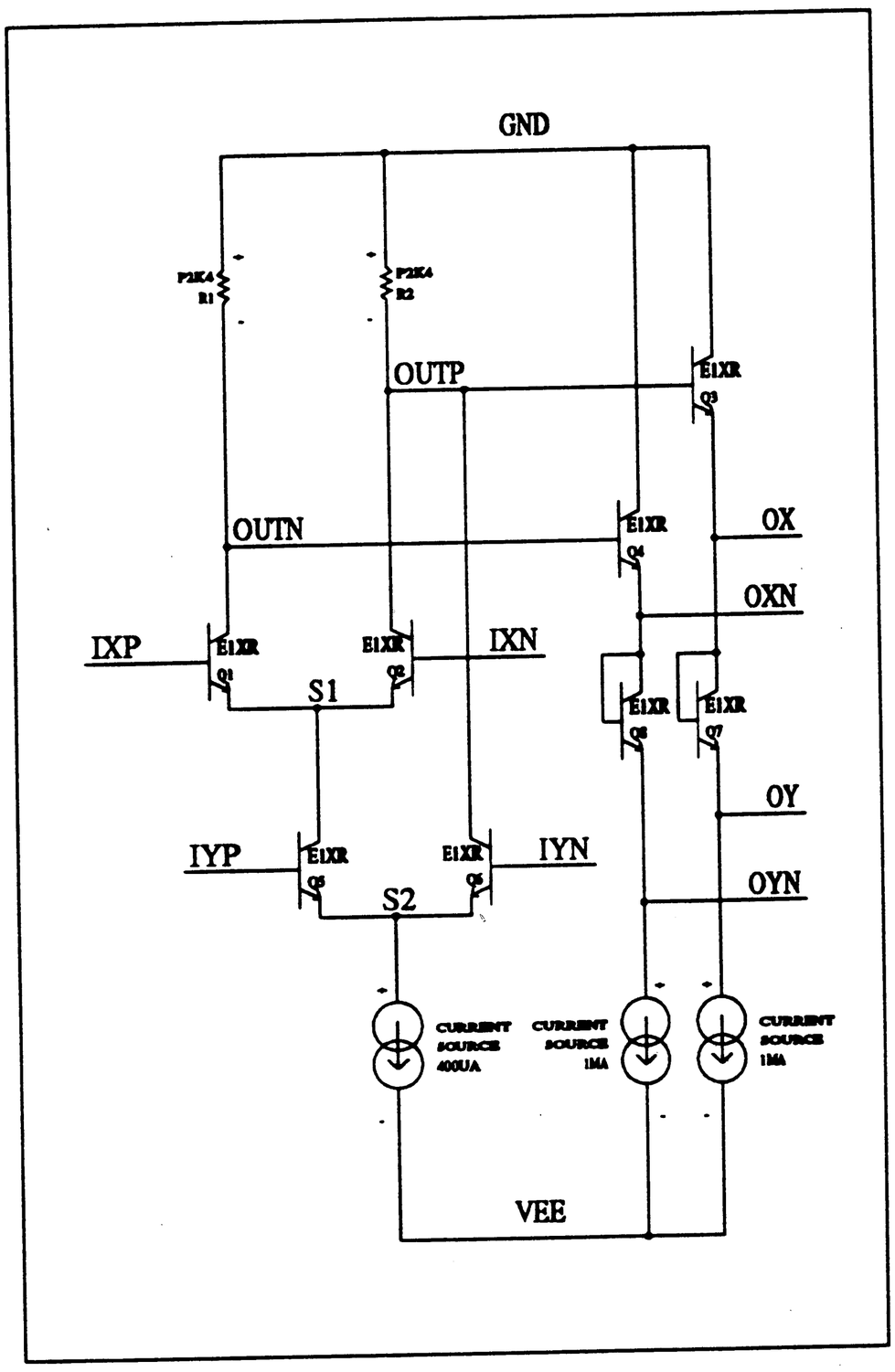


Figure #11. AND/NAND Gate with X and Y levels

5. Circuit Building Blocks

The background discussed in the last chapter sets the stage for the explanation of the seven internal building blocks and five input and output buffer blocks implemented in the design of the Time Division Multiplexer and Demultiplexer circuits. The naming of these building blocks grew out of the naming convention used for the function blocks in the AT&T BEST-1 Series High-Performance ECL Gate Arrays data sheet.[8] Rather than explain the naming convention, the following list is included as a look up table for the reader to explain the functionality of each of the building blocks.

RCR1HD - Single gate delay differential gate.

F3DAHD - Differential data latch with positive level enable.

F4DAHD - Differential Multiplexing data latch with positive level enable.

F1DAHD - Differential Master slave flip-flop.

F1DCHD - Differential Master slave flip-flop with asynchronous clear.

F1DCHD2 - Differential Master slave flip-flop with asynchronous clear.

F2DAHD - Differential Multiplexing Master slave flip-flop.

F2DCHD - Differential Multiplexing Master slave flip-flop with asynchronous clear.

F5DAHD - Three stage Master slave flip-flop.

EI1N1 - Single sided ECL data input buffer.

EBI1N1 - Differential ECL data input buffer.

EBCB - Differential ECL clock input buffer.

X1MN - 50Ω line driving Single-sided ECL output buffer.

XBMB - 50Ω line driving differential ECL output buffer.

First, the circuit in figure #12 is a differential ECL Inverter known as the RCR1HD. It operates identical to the previously discussed differential ECL Inverter in figure #9. This

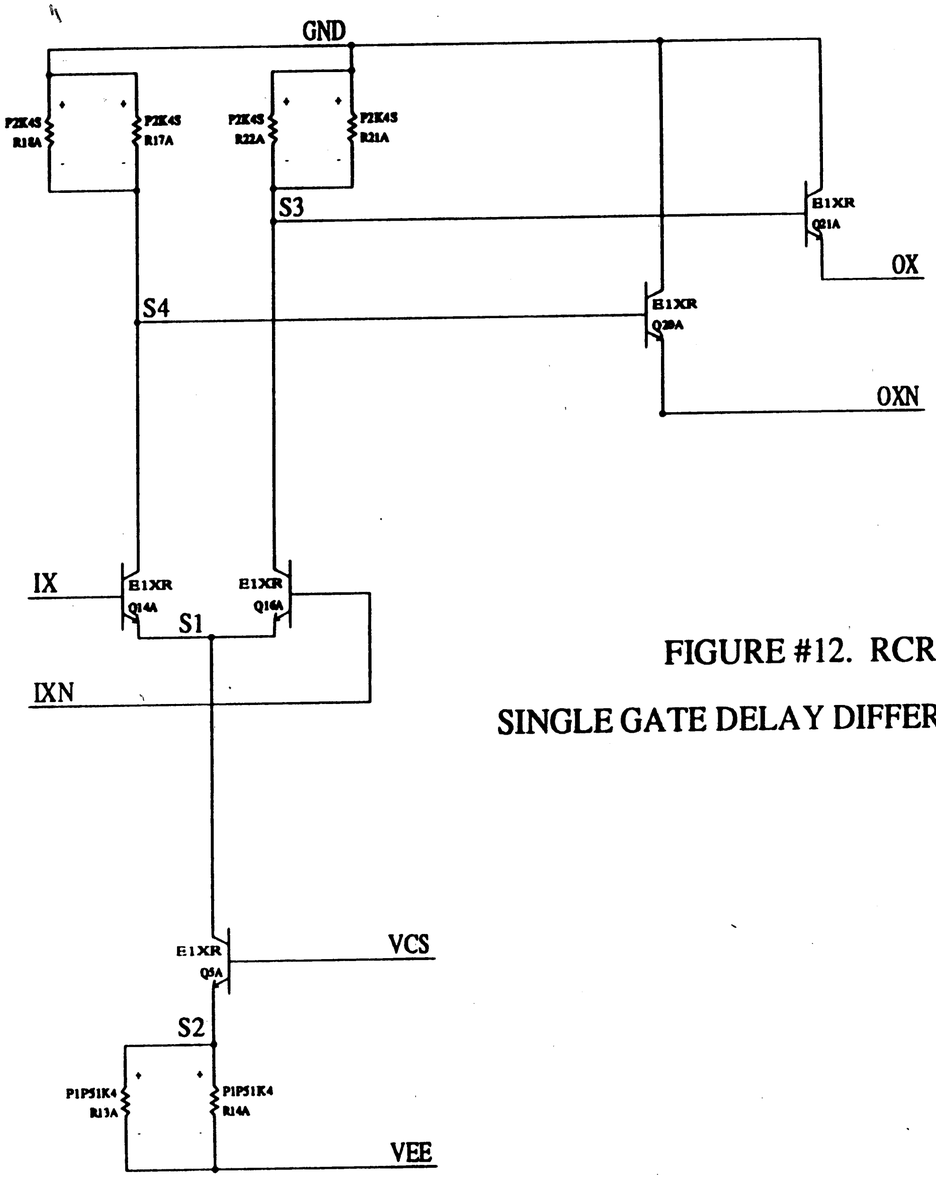


FIGURE #12. RCR1HD

SINGLE GATE DELAY DIFFERENTIAL GATE

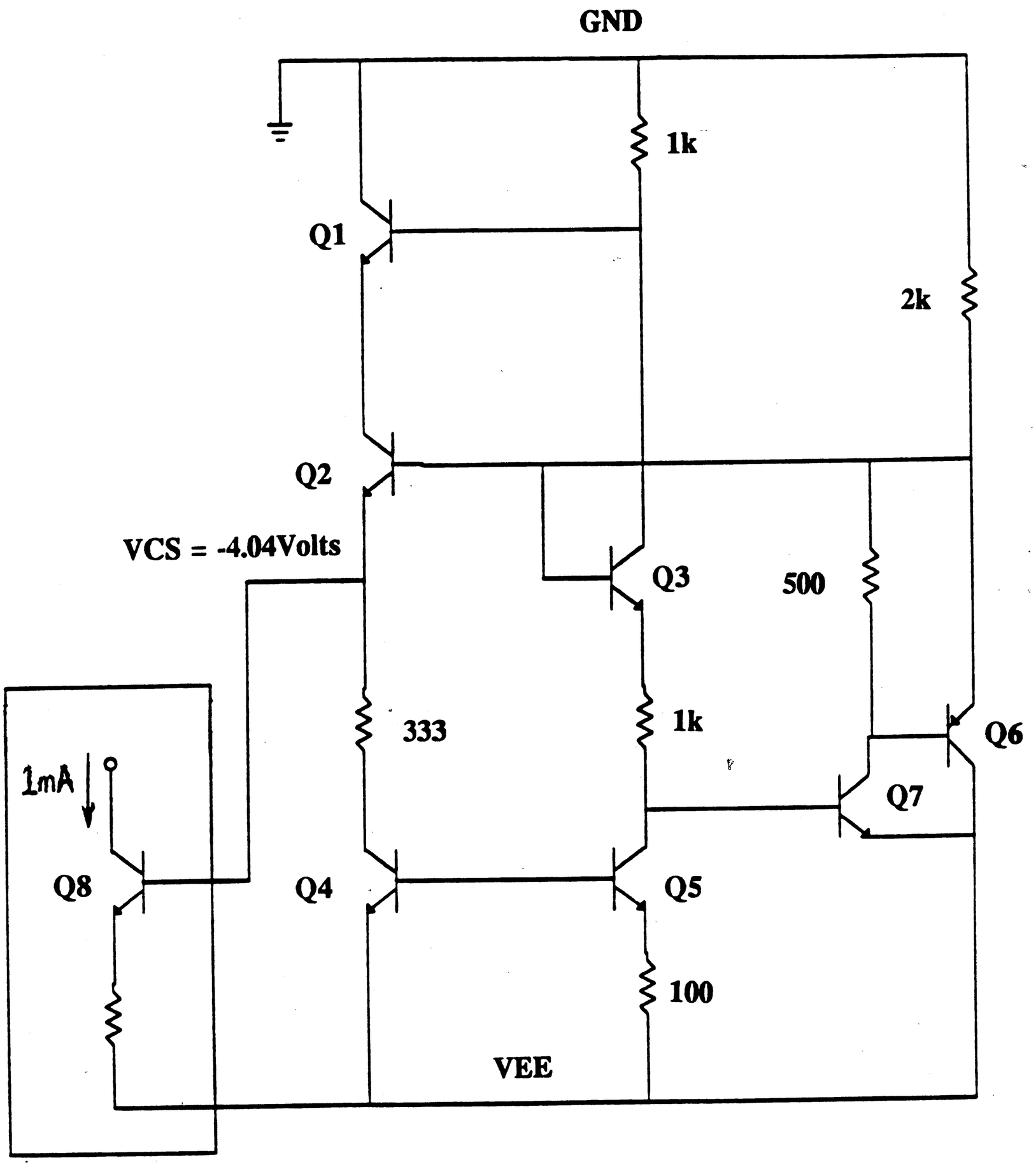


Figure #13 Band-Gap Reference Circuit

inverter does not produce the "Y" level outputs that were just discussed on the AND/NAND gate of figure #10, since the application for this gate in both circuits only requires the "X" level output. Figure #12 also shows the replacement of the 400μA constant current source symbol with transistor Q5A and two parallel 1.51kΩ emitter resistors R13A and R14A. The base of transistor Q5A is connected to a Band-Gap reference circuit,[9] shown in figure #13, which supplies a regulated voltage (VCS) of -4.05v which is temperature invariant.[10] This VCS voltage biases the transistor Q5A into its active region and tracks with variations of the V_{BE} of that transistor resulting in a controlled 300mV voltage across the parallel 1.51kΩ resistors (Vee equals -5.2 volts and the V_{BE} drop of 0.86 volts). Parallel resistors are used to have better control over the final value of the 755Ω resistor desired. If just one resistor were laid out, the resistor would be approximately one square of sheet resistance, and because of possible line width variations, it makes better sense to layout out two parallel resistors twice the size of your targeted resistor. Assuming infinite gain, the collector current of transistor Q5A will be equal to its emitter current of 400μA. Furthermore, the two 1milliamp (mA) constant current sources have been replaced by transistors Q1A and Q2A along with the 600Ω parallel resistor pairs R1A/R2A and R3A/R4A. The constant 300mV voltage drop across these resistor pairs results in each transistor supplying 1mA of load current to each of the outputs OX and OXN, resulting in a typical power dissipation for this building block of 12mW.

The next building block is the data latch F3DAHD shown in figure #14. The data latch controls the flow of data on a signal by either allowing the data to move transparently through the circuit, or storing the data in its latch and disabling the flow of further data. The input which controls this function is the "Y" level enable input EY/EYN. When EY is a logic high equal to -1.2 volts, transistor Q10A becomes active while transistor Q11A cuts-off. All of the tree current supplied by Q6A is then directed to the data input emitter coupled pair, Q13A and Q14A. Since transistor Q11A is cut-off, we can ignore the latch emitter coupled pair which consists of Q16A and Q15A. With just the data input emitter coupled pair active, the circuit closely resembles the differential ECL inverter described earlier in figure #9 where the DX input is the IXP input and DXN is the IXN input. When input EY goes to a logic low, the data no longer can move through the data latch transparently because transistor Q10A cuts-off and disables the data input emitter coupled pair by stopping the flow of tree current to node S2A. Now, Q11A becomes active and supplies the tree current to the latch emitter coupled

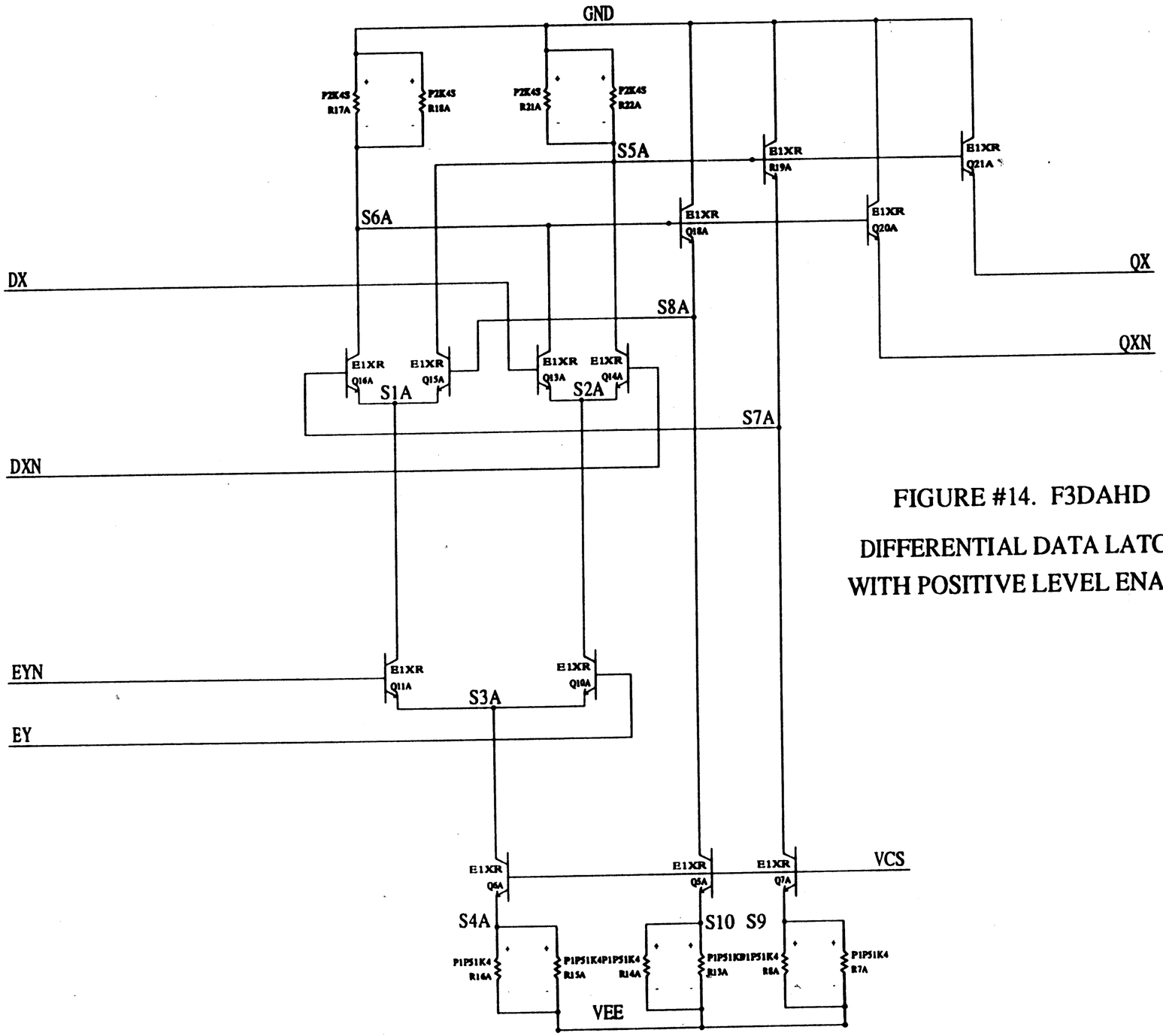


FIGURE #14. F3DAHD

DIFFERENTIAL DATA LATCH
WITH POSITIVE LEVEL ENABLE

pair at node S1A. The differential inputs to this emitter coupled pair are S7A and S8A which are equivalent in logic to outputs QX and QXN respectively. Looking further, it is apparent that the collectors of the latch emitter coupled pair are cross coupled such that the voltage levels on S7A and S8A will act to preserve their own logic states as well as the outputs. For example, when S7A is a logic high and its complementary signal S8A is a logic low, the latch emitter coupled pair will direct the tree current through resistors R17A and R18A and will result with a voltage on node S5A of 0.0 volts and -0.4 volts on node S6A. These voltages will then maintain the logic high on node S7A and the logic low on node S8A as well as the respective logic high on output OX and the logic low on OXN independent of what logic levels are on the inputs DX and DXN. In total, this circuit consumes approximately 16mW of power assuming 1mA output load current sources.

The multiplexing data latch F4DAHD is a slight variation of the above data latch and is the next building block to be examined. In figure #15, the circuit closely resembles the data latch except the single data emitter coupled pair is replaced by two parallel data emitter coupled pairs which are controlled by the stacked emitter coupled pair with inputs S0Y and S0YN. Furthermore, with the addition of this third level of logic, it becomes necessary to level shift the EY and EYN signals down a V_{BE} in order to protect transistor Q12A from becoming saturated. At this point, one might question whether it would be beneficial to add another diode in all of the outputs to create a "Z" level of logic. But, this level is not as prevalent in all of the building blocks as the "Y" level, therefore the benefits of having the extra level will not cover the loss in output switching performance do to the additional capacitance of the diode. As with the data latch F3DAHD, when the EY input is a logic high, the transistor Q12A becomes active and directs the tree current to select the emitter coupled pair Q9A and Q10A. If S0Y is a logic high, Q9A will become active and will direct the tree current to the D1X/D1XN data emitter coupled pair. However, if S0Y is a logic low, Q10A will become active and will direct the tree current to the D0X/D0XN data emitter coupled pair. With either case, once the tree current has been directed to the specific data emitter coupled pair, the data will move transparently to the output as explained earlier for the F3DAHD data latch. When EY goes to a logic low, the multiplexing latch will hold whatever logic levels are located on the QX and QXN outputs upon biasing the emitter couple latch Q14B and Q15B. The additional level shifting current which is needed in this circuit over the F3DAHD latch

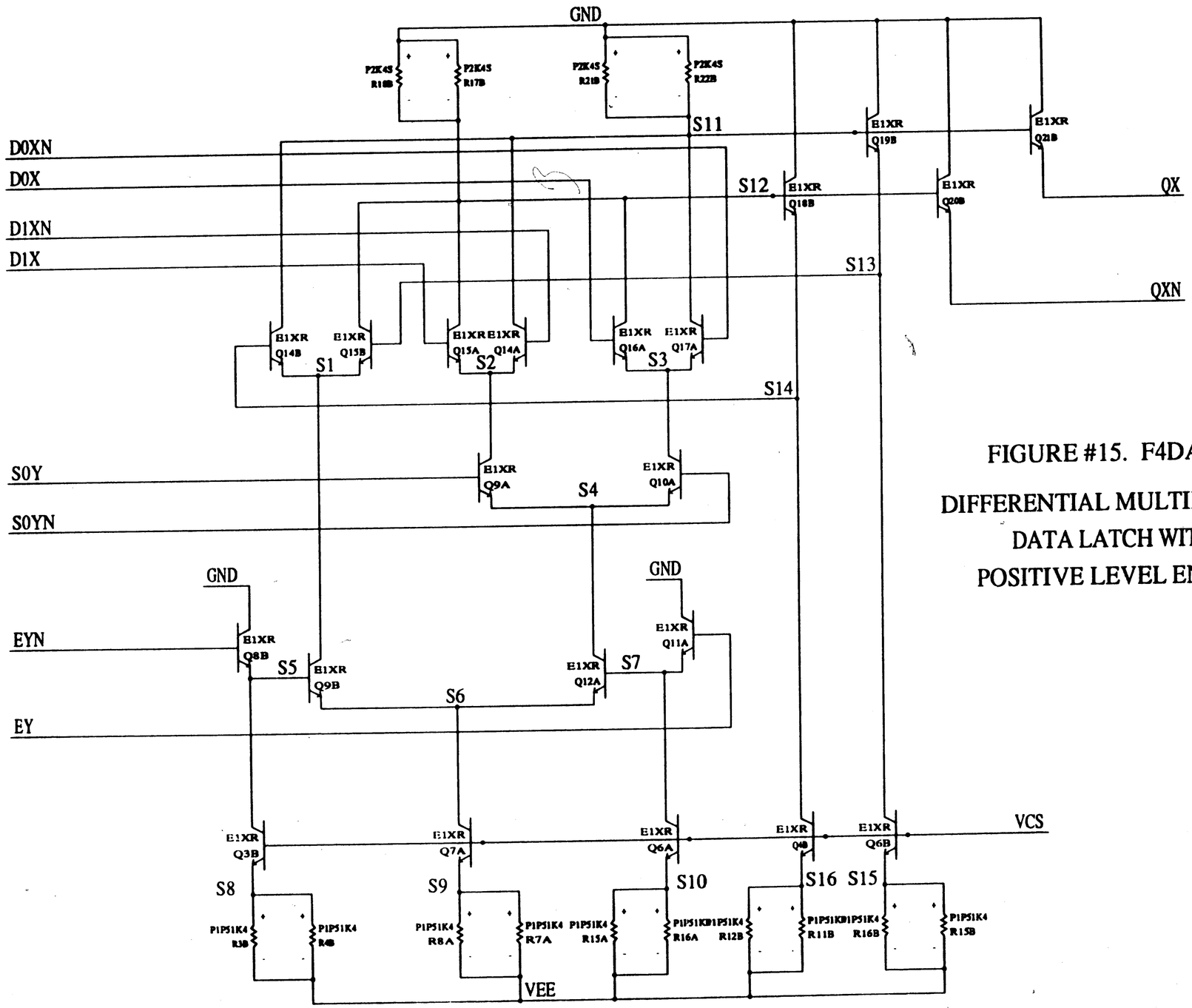


FIGURE #15. F4DAHD

DIFFERENTIAL MULTIPLEXING
DATA LATCH WITH
POSITIVE LEVEL ENABLE

increases the power consumption of this latch to 20mW including the 1mA emitter follower load currents on the outputs OX and OXN.

The next building block is the positive edge triggered Master-slave D-type flip-flop F1DAHD which consists of two cascaded data latches as shown in figure #16. The first stage of the D-type flip-flop consists of a F3DAHD with its EYN input connected to the CKY input and its EY input connected to the CKYN inputs. This latch is known as the Master latch and its major function is to acquire new data on the inputs DX and DXN while the CKY input is a logic low. Then, when the CKY input goes high, it latches this data and holds it stable for the next stage of the circuit. This second stage which also consists of a F3DAHD, has its EYN input connected to the CKYN input and its EY input connected to the CKY input. Therefore, when the master latch is acquiring new data, the second latch (which is known as the slave latch) is in an output latched state holding the old data stable, while also inhibiting the transparent flow of the new data to the outputs QX and QXN. Later, when the master latch becomes latched and presents new stable data, the slave latch becomes transparent and enables the new data to propagate to the outputs QX and QXN. The circuit is considered to be positive edge triggered because once there is a positive logic edge on the input CKY, the present data at the input is transferred and held stable at the output, while the master data latch disables the input data emitter coupled pair and inhibits any further data changes on DX/DXN to enter the gate. This circuit consumes 22mW including the 1mA emitter follower currents for outputs QX and QXN.

With the addition of two RCR1HD differential inverters to the F1DAHD Master-slave D-type flip-flop, the F1DCHD and F1DCHD2 Master-slave D-type flip-flops with asynchronous clear can be implemented. As shown in figures #17 and #18, a RCR1HD differential inverter with input IXP controlled by CLX and input IXN controlled by CLXN has been added to each latch of the D-type flip-flop. When the CLX input is a logic low, transistors Q8A and Q13B are cut-off and transistors Q9A and Q17A are active. This directs the tree currents from Q4A and Q4B through the load resistors R20A/R19A and R20B/R19B pulling nodes S21 and S24 to logic lows of -0.4 volts. The emitter follower transistors Q18A and Q19A are connected in a wired OR configuration with transistors Q20A and Q18B respectively. In the wired OR configuration which ever emitter followers base voltage is higher, that emitter follower will override the other and will control the logic on the common emitter output node.

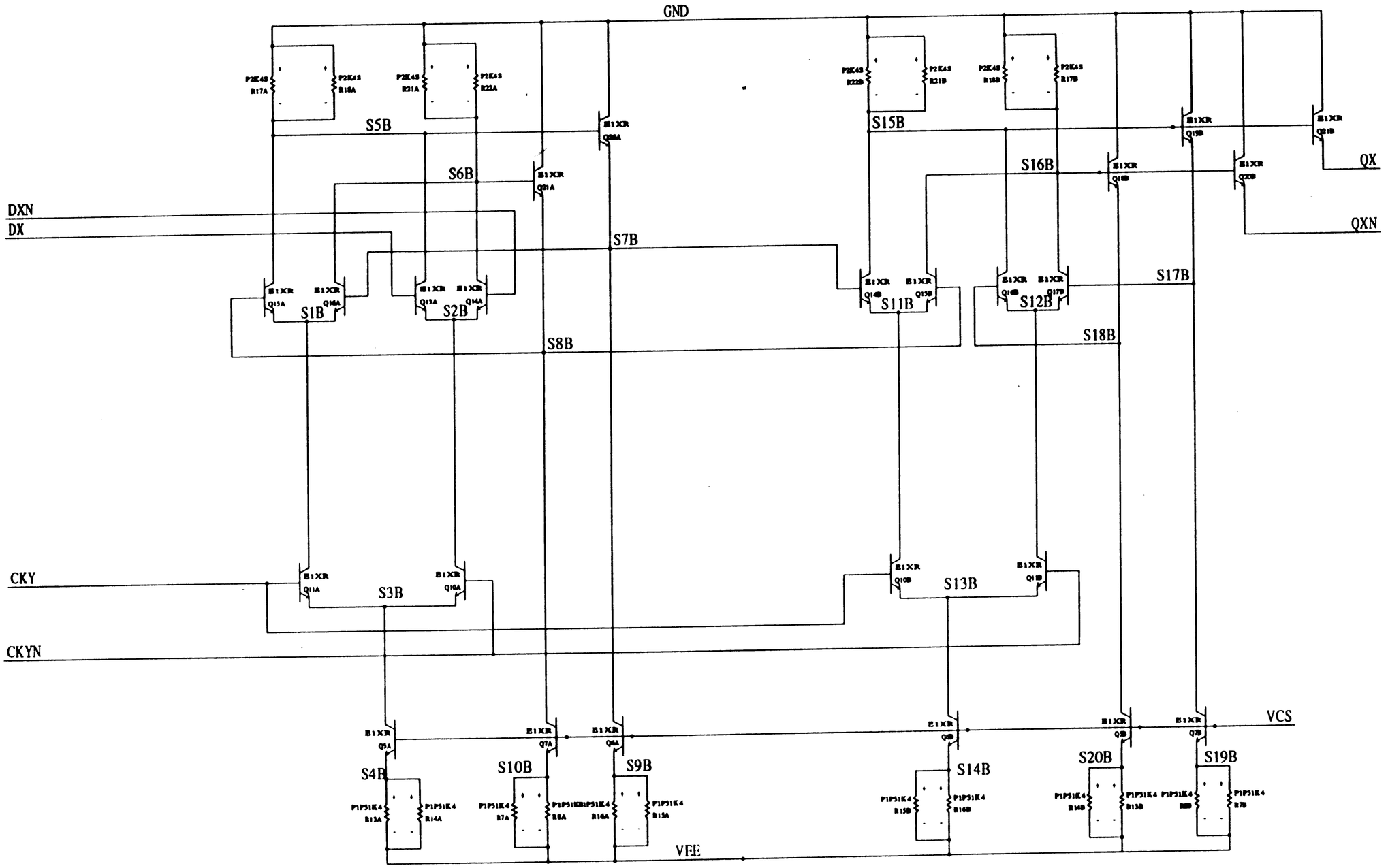


FIGURE #16. FIDAHD - DIFFERENTIAL MASTER SLAVE FLIP-FLOP

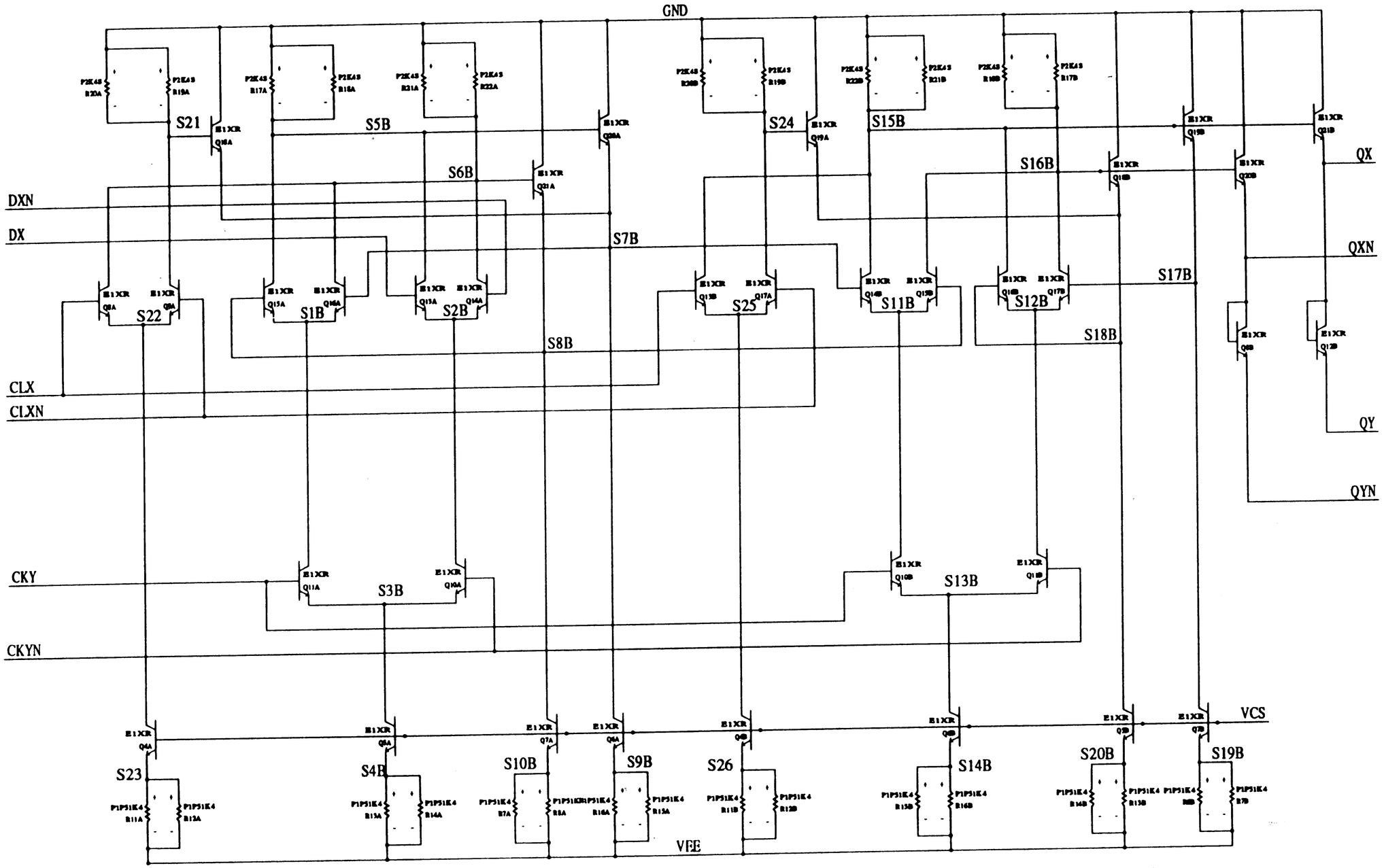


FIGURE #17. F1DCHD- DIFFERENTIAL MASTER SLAVE FLIP-FLOP WITH ASYNCHRONOUS CLEAR

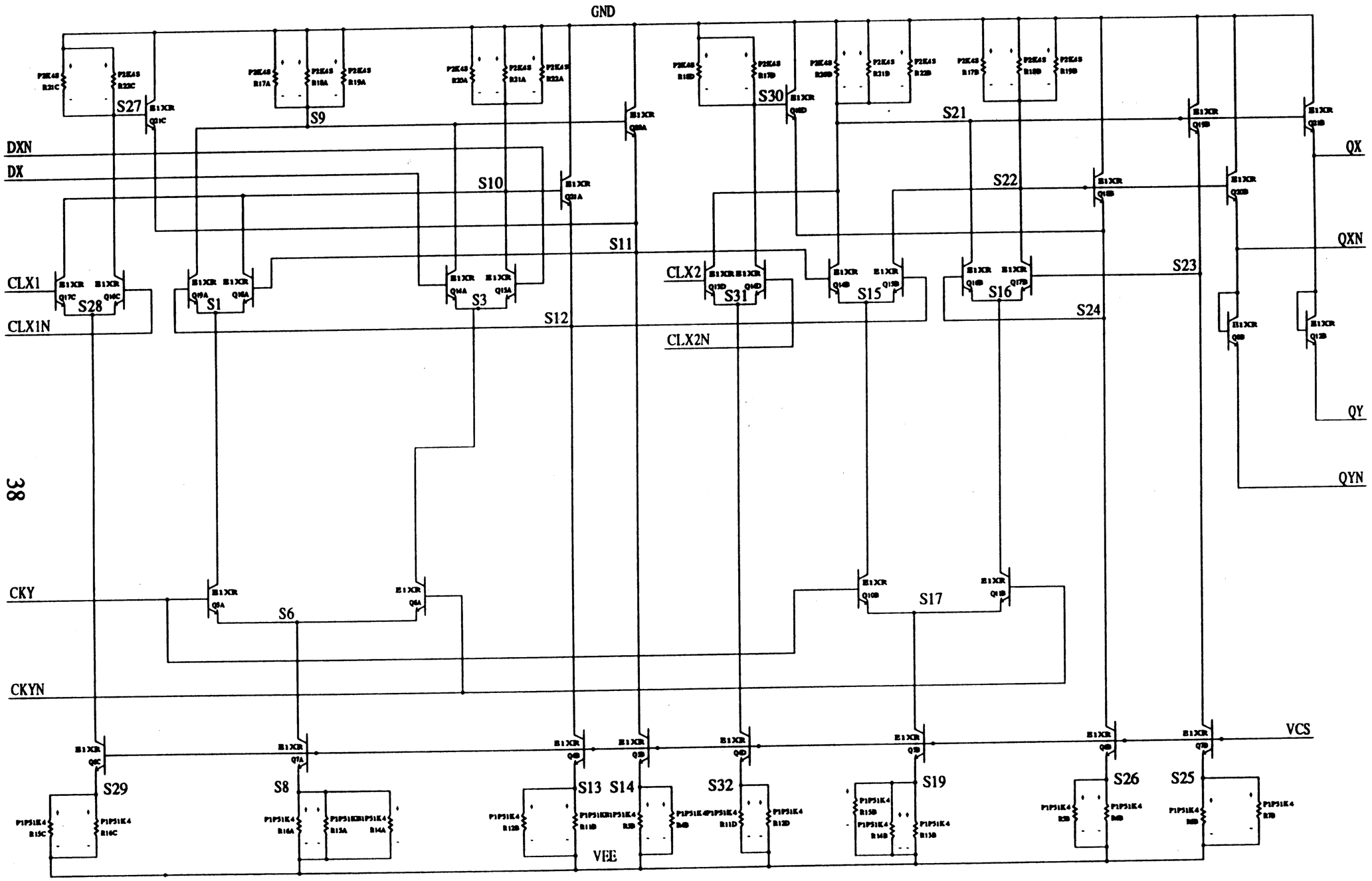


FIGURE #18. F1DCHD2 - DIFFERENTIAL MASTER SLAVE FLIP-FLOP WITH ASYNCHRONOUS CLEAR

Therefore, with nodes S21 and S24 as logic lows, the transistors Q20A and Q19B will always control the logic on the nodes S7B and S18B. However, when the CLX input is a logic high, transistors Q8A and Q13B become active while transistors Q9A and Q17A cut-off. With transistors Q8A and Q13B active, tree currents are directed through resistors R21A/R22A and R21B and R22B which pull nodes S6B, S8B, S15B, and S17B to logic lows. At the same time, node S21 and S24 become logic highs of 0.0 volts and override the emitter follower transistors Q20A and Q18B and pull the nodes S7B and S18B to logic highs. The final result is the master latch is cleared and the OX output which is node S7B becomes a logic low and the OXN output which is node S8B becomes a logic high. Furthermore, the slave latch is also cleared and the QX/QY outputs become logic lows and the QXN/QYN outputs become logic highs independent of whatever is happening on the CKY/CKYN inputs.

One potential problem does exist while clearing the flip-flop when the CKY input is a logic low and the data input DX is a logic high. This will bias transistors Q14A and Q10A active and will direct the tree current from transistor Q5A through resistors R21A/R22A. But, it was just shown that when the CLX input is a logic high and the master latch is being cleared, the tree current from transistor Q4A also gets directed through the resistors R21A/R22A. With two 400μA tree currents being directed through these resistors, the node voltage on S6B becomes -0.8 volts and node S8B goes to -1.6 volts. This presents a possible saturation condition for transistor Q10B. To avoid this problem, the clearing of this D-type flip-flop will only be done when the CKY input is a logic low and the DX input is a logic low. This removes all possibilities of two tree currents being directed through any single current swing resistor.

The F1DCHD and F1DCHD2 D-type flip-flops with clears are identical in logic, but have a minor difference in tree current. The F1DCHD2 parallels three emitter resistors in the tree current sources to obtain 600μA rather that 400μA. This extra tree current is needed to obtain higher switching speeds, since this flip-flop is used in the first stage of the ripple counter of the Time Division Multiplexer and needs to operate twice as fast as the other F1DCHD flip-flops in the counter. Upon increasing the tree current to 600μA, the load resistors are also lowered to 667Ω in order to maintain the internal 400mV signal swing. The typical power dissipation for each of these flip-flops is 26mW for the F1DCHD and 28mW for the F1DCHD2 flip-flop including the 1mA current sources for the output nodes.

By replacing the F3DAHD master latch with a F4DAHD data latch in the F1DAHD, the F2DAHD positive edge triggered Mux-type flip-flop can be realized as shown in figure #19. Note in the schematic though, that the placement of the F4DAHD latch requires the input clock CKY/CKYN to be shifted down V_{BE} in logic level. In order for this logic level to control the slave latch, the F3DAHD slave latch must also have its CKY/CKYN input clock signals lowered. Despite this, the logic of this circuit is very similar to that of the F1DAHD. When the CKY input is a logic low, the Mux-type master data latch will acquire new data from inputs D0X/D0XN or D1X/D1XN depending on the state of S0Y/S0YN. Meanwhile, the slave latch will be latched with the old data and will inhibit the transparent flow of the new data to the outputs QX/QXN. When the CKY input goes to a logic high, the Mux-type master data latch will capture the new data and will hold it stable while the slave latch moves into its transparent state and allows the new data to propagate to the outputs. As with the F1DAHD, once an active positive clock edge occurs on into CKY, further data changes on inputs DX/DXN will be inhibited from the gate because the input emitter coupled pair stages will be cut-off. As with the F1DCHD2, this flip-flop requires high switching speeds. Therefore, its tree currents are increased to 600μA, yielding a typical power consumption for this gate of 28mW including 1mA load currents on the output nodes.

As with the F1DCHD and F1DCHD2, upon adding two RCR1HD differential inverters to the F2DAHD Mux-type flip-flop, the F2DCHD positive edge triggered Mux-type flip-flop with asynchronous clear is implemented as shown in figure #20. The addition of this differential inverters is done the same as in the F1DCHD and F1DCHD2 flip-flops, and they perform the same with a logic high on CLX in conjunction with a logic low on CLXN to clear both the master and slave latches and pull the QX/QY outputs to logic lows and the QXN/QYN outputs to logic highs. The problem with dual tree currents also exists with this flip-flop. Therefore, the clearing of this gate will be done only when the CKY input and D0X/D1X data inputs are logic lows. The typical power consumption for this gate is 32mW including the 1mA load currents on the output nodes.

The last internal building block is the F5DAHD Three stage flip-flop[1] which consists of a cascade of a F3DAHD data latch and a F1DAHD D-type flip-flop as shown in figure #21. This flip-flop is used in the demultiplexing stages of the Time Division Demultiplexer and its application as will be seen in chapter 7, has its EY and CKY inputs connected and its EYN

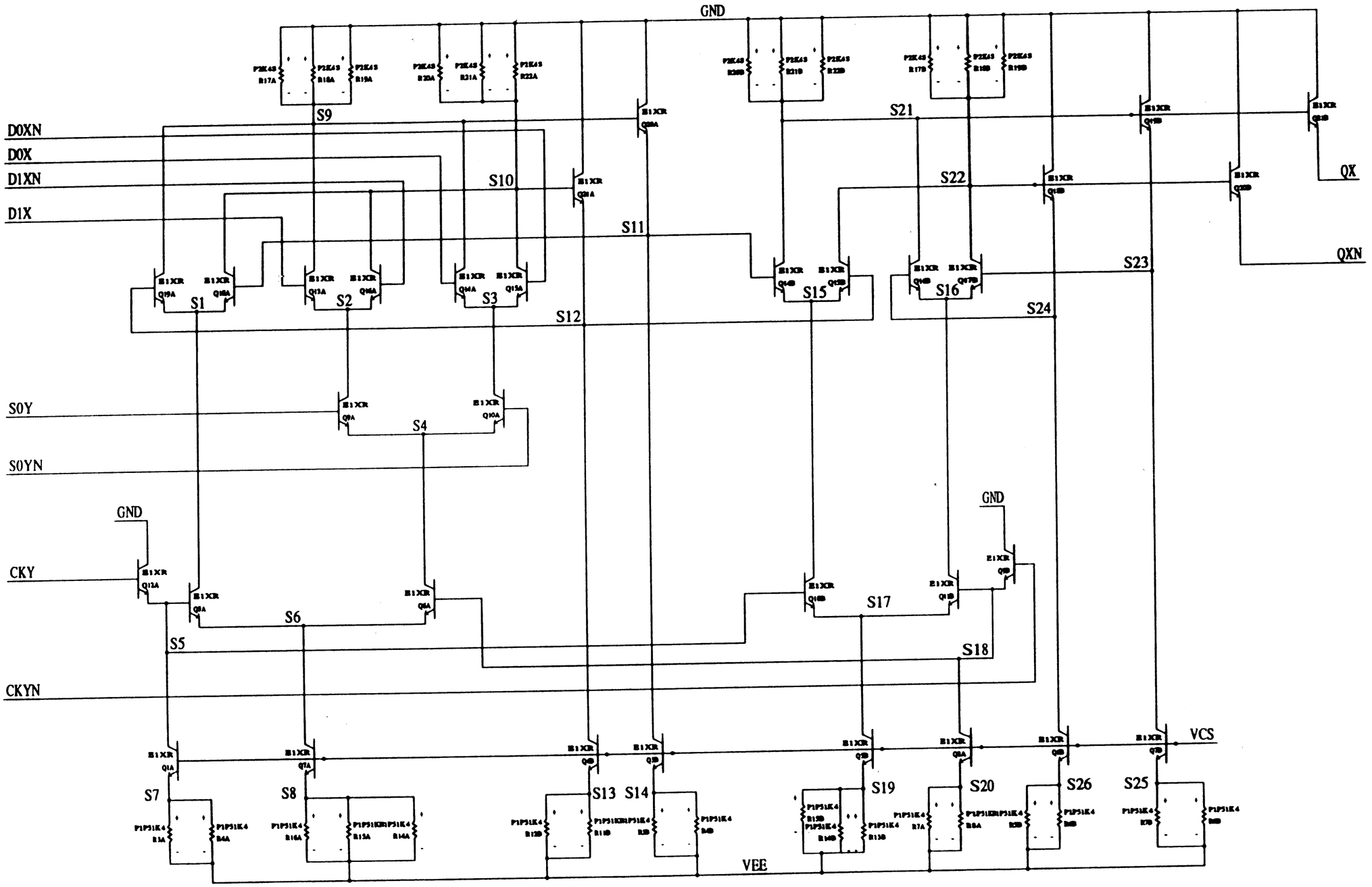


FIGURE #19. F2DAHD - DIFFERENTIAL MULTIPLEXING MASTER SLAVE FLIP-FLOP

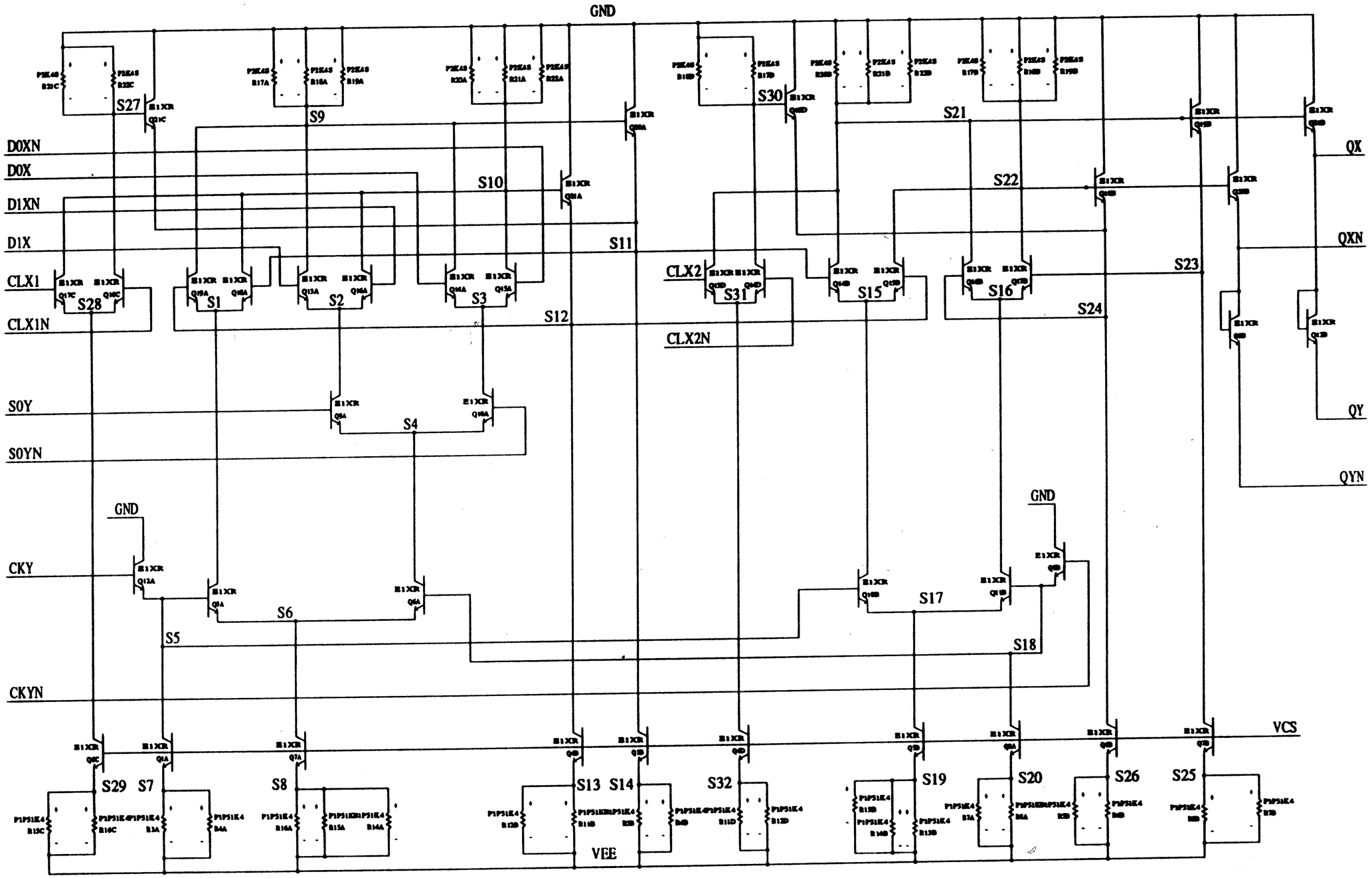


FIGURE #20. F2DCHD - DIFFERENTIAL MULTIPLEXING MASTER SLAVE FLIP-FLOP WITH ASYNCHRONOUS CLEAR

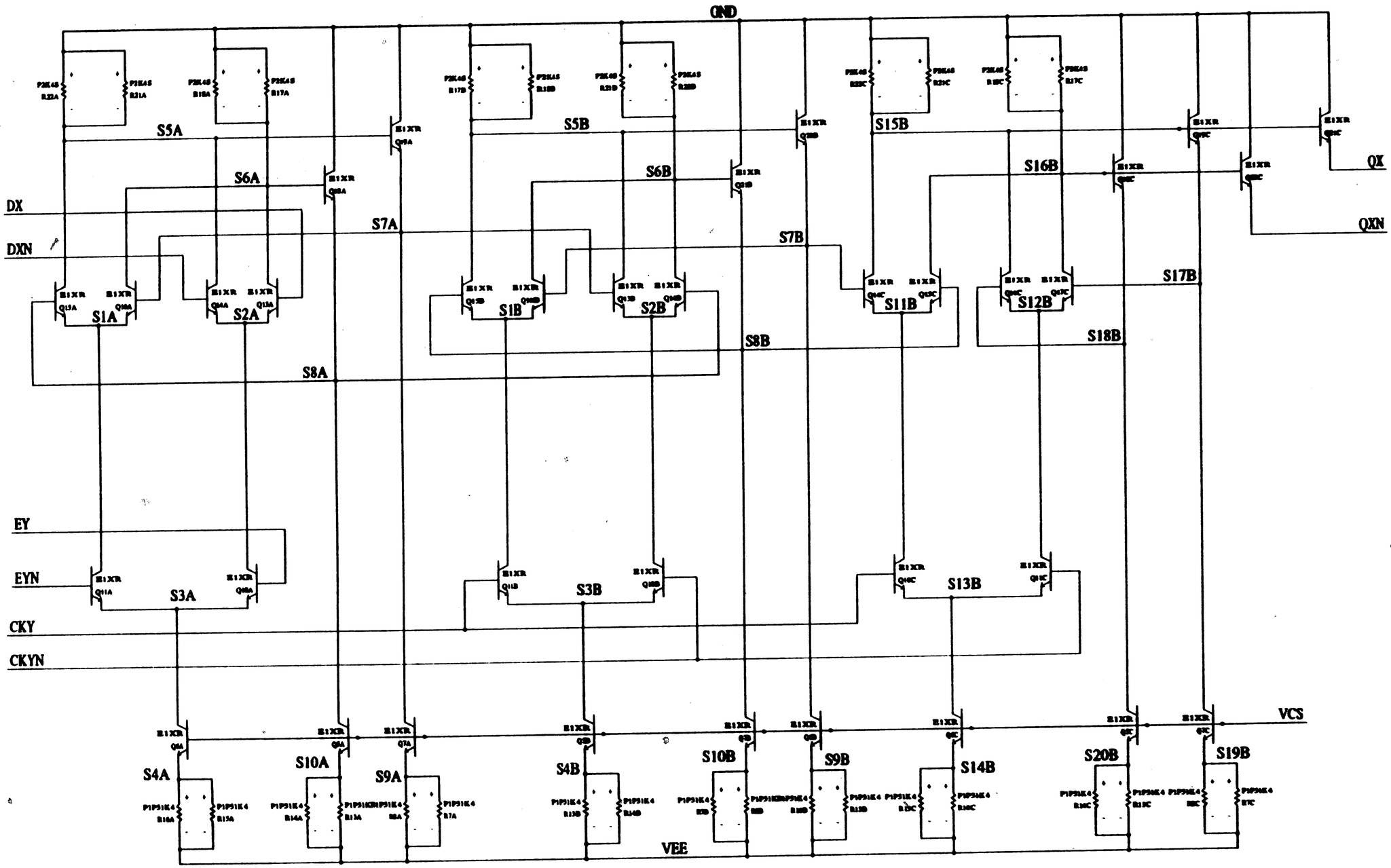


FIGURE #21. F5DAHD - THREE STAGE MASTER SLAVE FLIP-FLOP

and CKYN inputs connected. Therefore, when the EY/CKY input is a logic high, the first stage data latch acquires new data while the first stage of the F1DAHD is latched and inhibits this new data from effecting the outputs QX and QXN. As the EY/CKY input has a negative falling edge, the F3DAHD latches whatever data it had acquired and holds it stable while the first stage of the F1DAHD moves into an acquire mode. Meanwhile, the last stage of the F1DAHD is latched and inhibits the transparent flow of this new data to the outputs. When the EY/CKY input has its next rising edge, the first stage of the F1DAHD holds the new data that it acquired from the F3DAHD data latch while the second stage becomes transparent and enables this new data to propagate to the outputs QX and QXN. Furthermore, once the EY/CKY input completes its rising edge, the F3DAHD data latch begins acquiring new data. On a macro level, this flip-flop effectively is a negative edge triggered flip-flop that delays the delivery of its captured data until the following rising edge clock edge. The typical power consumption for this gate is 21mW including 300μA load current sources on the output nodes instead of 1mA current sources like the other internal building blocks.

In total, there are three ECL level input buffers and two ECL output buffers. The EI1N1 input buffer as shown in figure #22, functions as a single sided ECL inverter as described earlier in figure #9, and it consumes 9mW of power assuming 500μA of load current on each of the output nodes. EBI1N1 as shown in figure #23, functions as a RCR1HD differential ECL inverter as described earlier in figure #12, and it consumes 12mW of power with 1mA current sources on each of its output nodes. The last of the input buffers is the EBCB differential clock driver as shown in figure #24. This buffer also functions as the RCR1HD. However, it is designed to drive high speed internal clock nodes and therefore it consumes 60mW of power including the 2mA of load current for each output node. The output buffer X1MN is a differential non-inverting output buffer as shown in figure #25. It functions just like the RCR1HD except the inverting side of the circuit has been eliminated. This buffer consumes 20mW of power. The last output buffer is the XBMB and is shown in figure #26. It also functions just like the RCR1HD except the IXP and IXN inputs are level shifted through a common collector amplifier to provide a current gain stage to drive the high speed large transistor emitter coupled pair of Q16A and Q16B. This current gain stage also helps maintain a 50% switching duty cycle at the outputs assuming that the inputs are a controlled 50% switching duty cycle. The necessity for this will become more apparent in chapter 6 when the

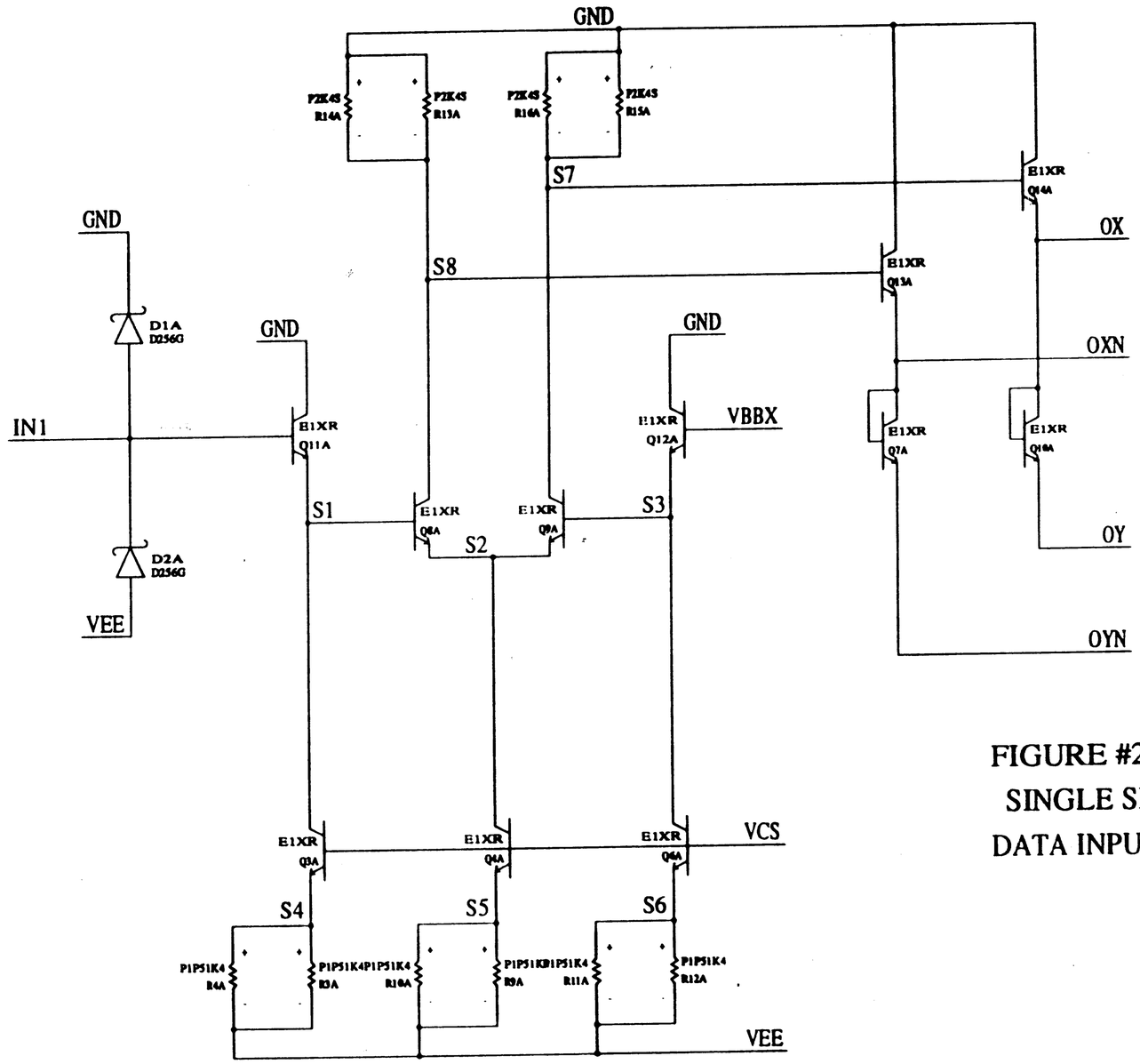


FIGURE #22. EI1N1
SINGLE SIDE ECL
DATA INPUT BUFFER

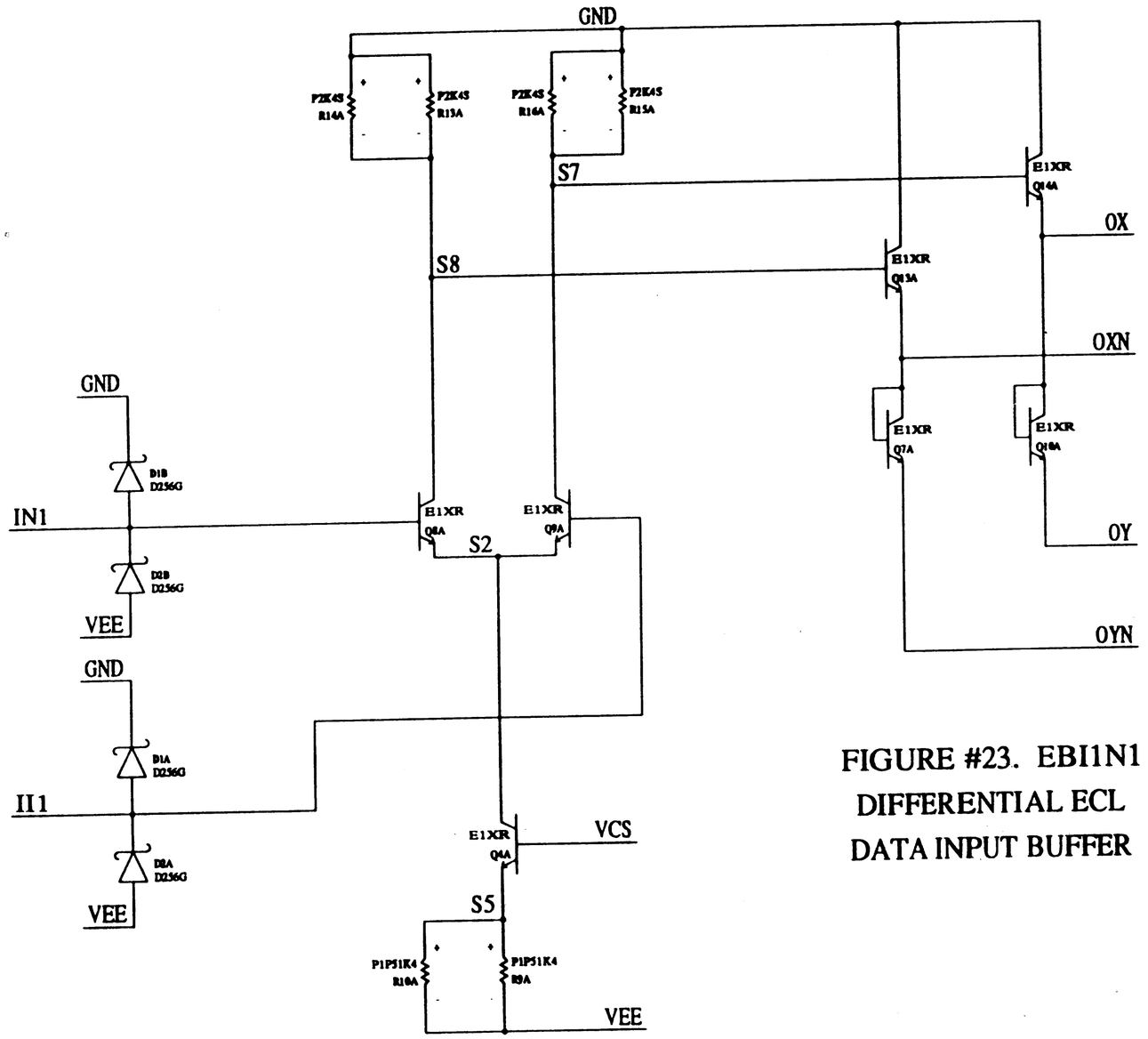


FIGURE #23. EBI1N1
DIFFERENTIAL ECL
DATA INPUT BUFFER

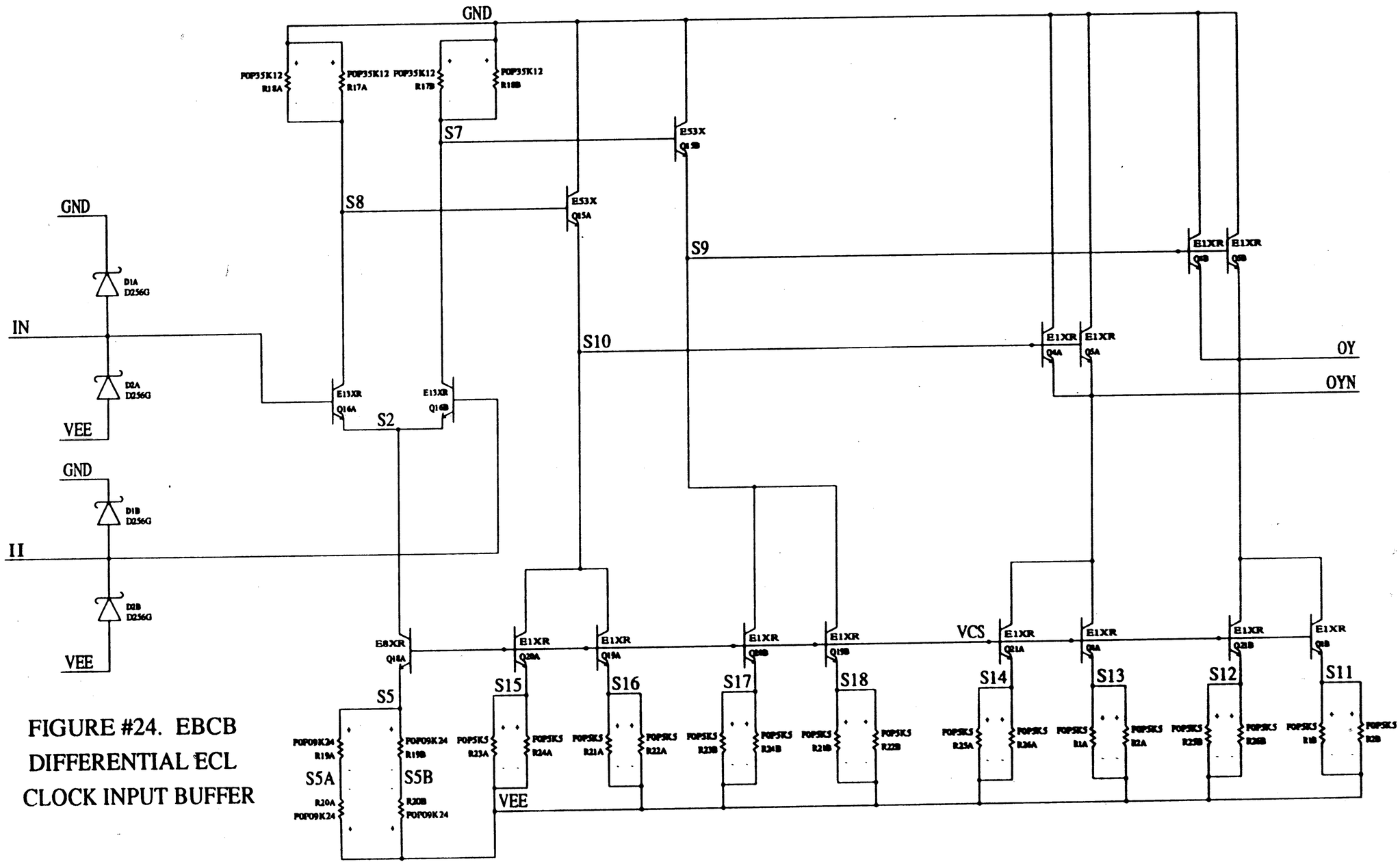


FIGURE #24. EBCB
DIFFERENTIAL ECL
CLOCK INPUT BUFFER

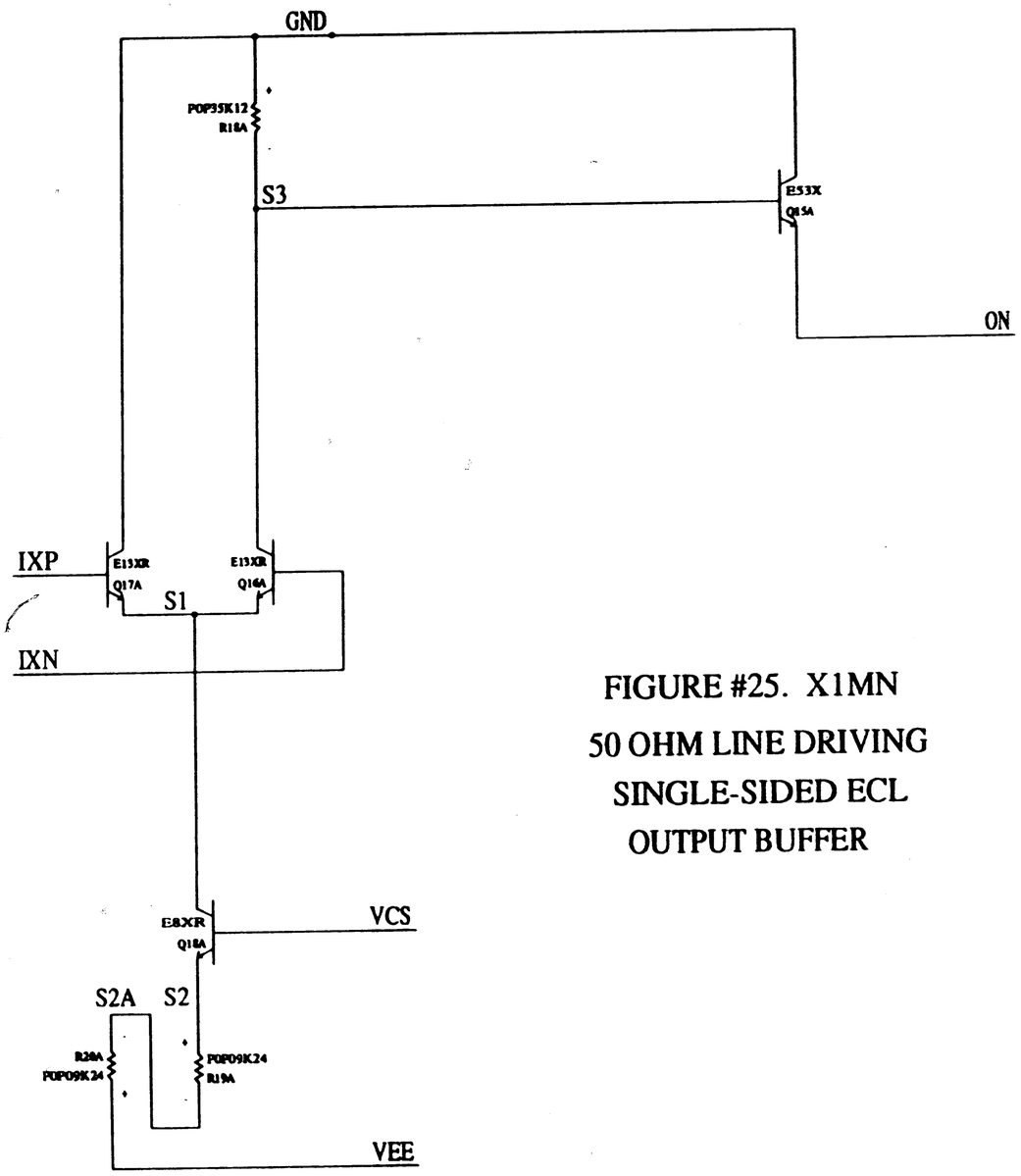


FIGURE #25. X1MN

50 OHM LINE DRIVING

SINGLE-SIDED ECL

OUTPUT BUFFER

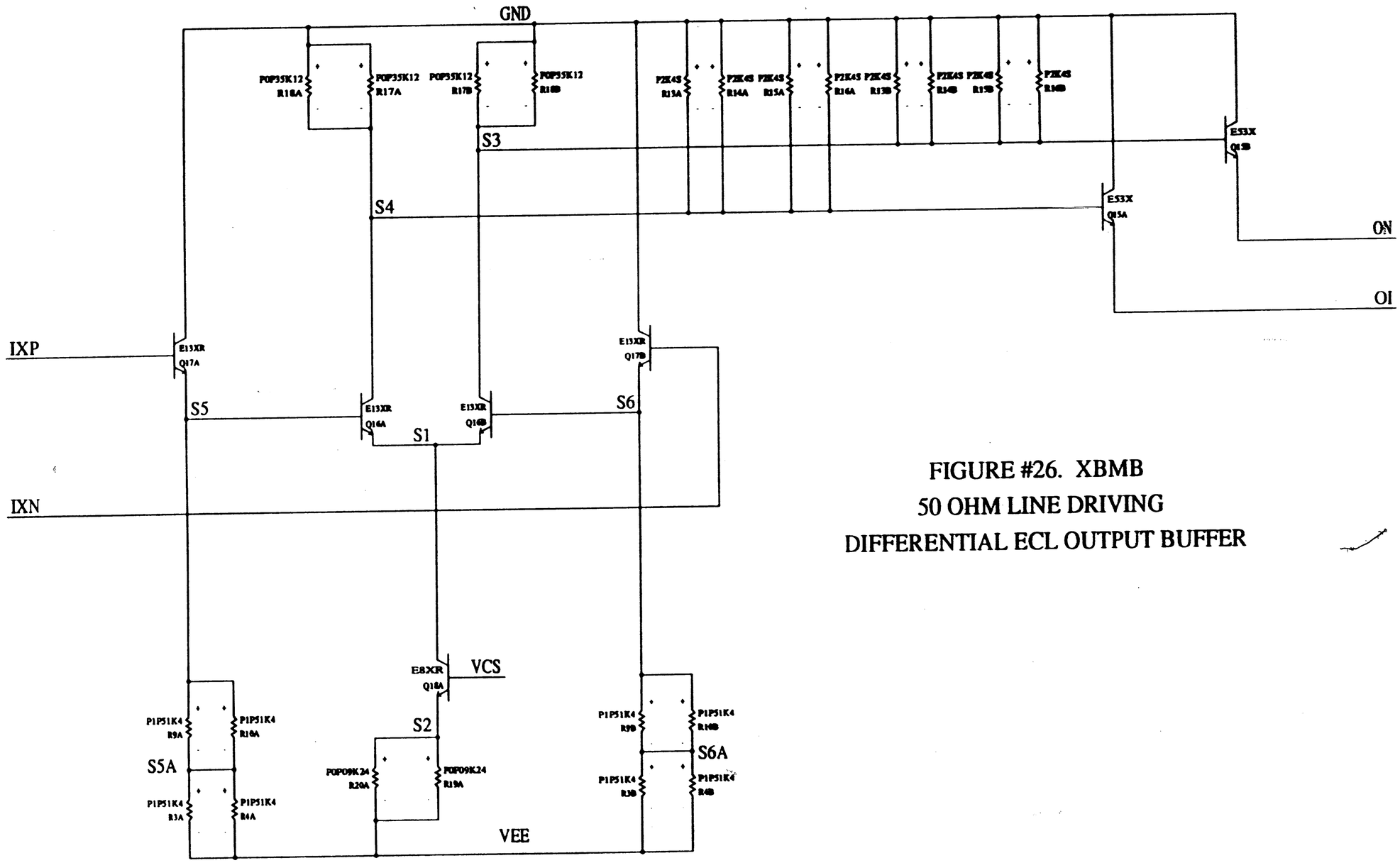


FIGURE #26. XBMB
50 OHM LINE DRIVING
DIFFERENTIAL ECL OUTPUT BUFFER

jitter results are analyzed for the Time Division Multiplexers output. The power consumption for this buffer is 67mW.

6. 8 Channel to 1 Channel Bit Interleaving Time Division Multiplexer

The 8 Channel to 1 Channel Bit Interleaving Time Division Multiplexer is a full differential ECL circuit with 400mV internal logic signal swings. It combines eight 300MHz data channels into a single 2.5Gbit/s continuous serial data bit stream and consists of 11 inputs and 6 outputs as shown in figure #27. Eight of the inputs are the data channels MDATA0-MDATA7. The multiplexing of these input channels is controlled by a 3 bit ripple counter shown in figure #28, which is clocked by the input 2.5GHz differential sinusoidal input clock CLKPM and CLKNM. For testing purposes, input CLRM is required to clear the ripple counter and get the internal signals MCKY2P, MCKY4P, and MCKY8P into known states. The multiplexed data produced by the circuit is transmitted from the 50Ω differential outputs OUTP and OUTN. Each new word of information transmitted is framed by the 50Ω differential frame clock output from FRPM and FRNM. Furthermore, the circuit also supplies a differential version of the input 2.5GHz clock which can be transmitted along with the output data stream from the TRPM and TRNM differential outputs.

To avoid confusion, the analysis of this circuit will be done with reference to the positive versions of the differential signals since these are defined to be complementary. To begin, lets look at the 3 bit ripple counter which is the nerve center of the design. The counter consists of a cascade of three flip-flops with clears (F1DCHD, F1DCHD2), the Q outputs of which are set to logic zeros by a positive voltage pulse applied to the external CLRM input. Since each D flip-flop has its true data input connected to its complementary output (i.e Toggle mode), each positive voltage transition on the CKY inputs will cause the D flip-flops to toggle their Q outputs into an opposite logic state. Therefore, after the clearing of the counter, the first positive voltage transition at the input CKPM will cause the D flip-flop MFF1 to toggle, resulting in CKY2PM to move from a logic low state to a logic high state. This positive transition will then trigger the D flip-flop MFF2 to toggle sending its Q output CKY4PM into a logic high state. Finally, the positive voltage transition of CKY4PM will toggle the D flip-flop MFF3 which will cause the Q output of the last stage CKY8PM to go to a logic high state. In total, the single clock edge from CKYPM causes the counter to move from a binary count of 0 (000) to a count of 7 (111). Furthermore, with every positive transition with CKYPM after this time, the MFF1 flip-flop output will toggle producing a clock signal of 1.25GHz on signal CKY2PM at half the frequency of the 2.5GHz CKYPM input clock.

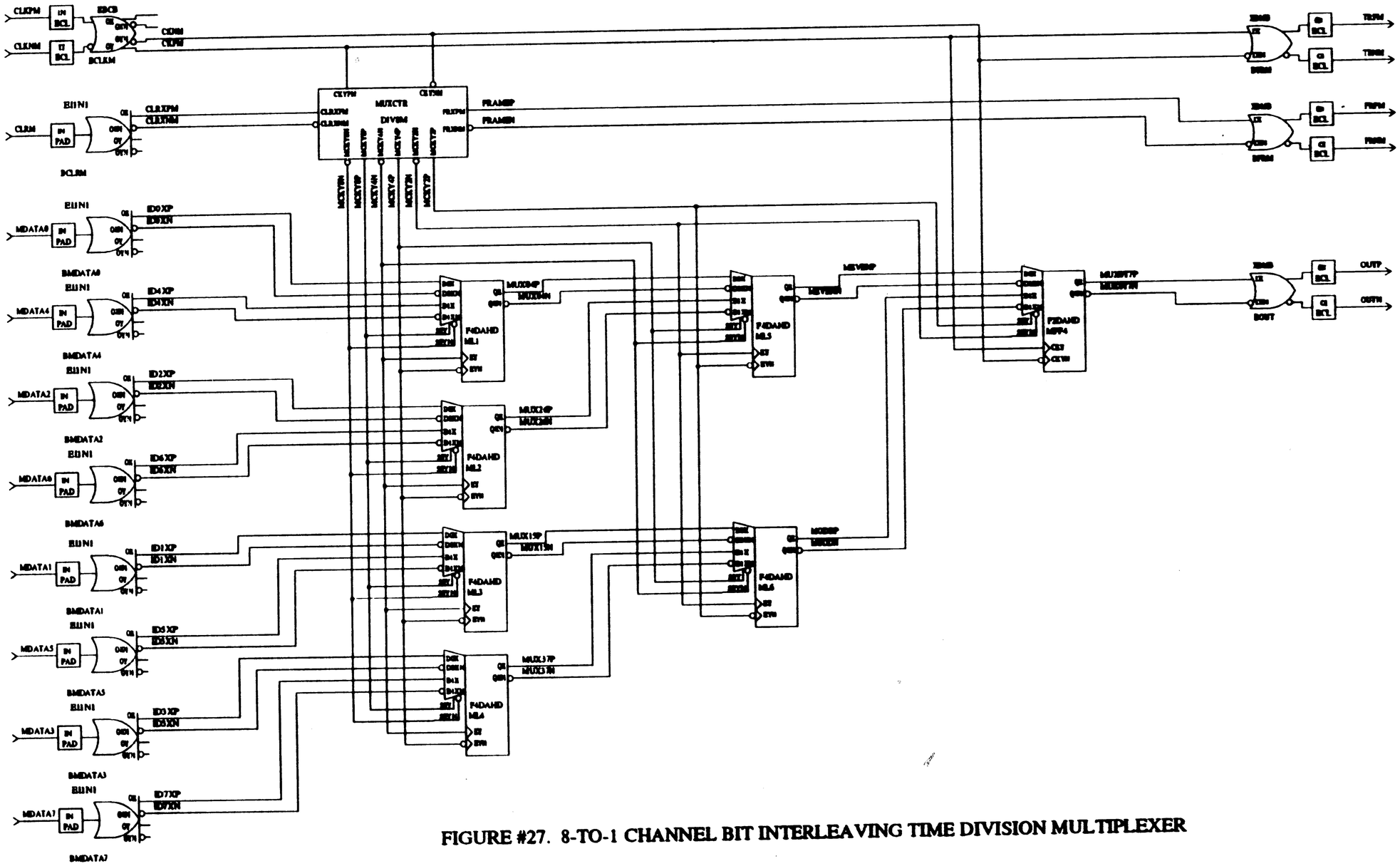


FIGURE #27. 8-TO-1 CHANNEL BIT INTERLEAVING TIME DIVISION MULTIPLEXER

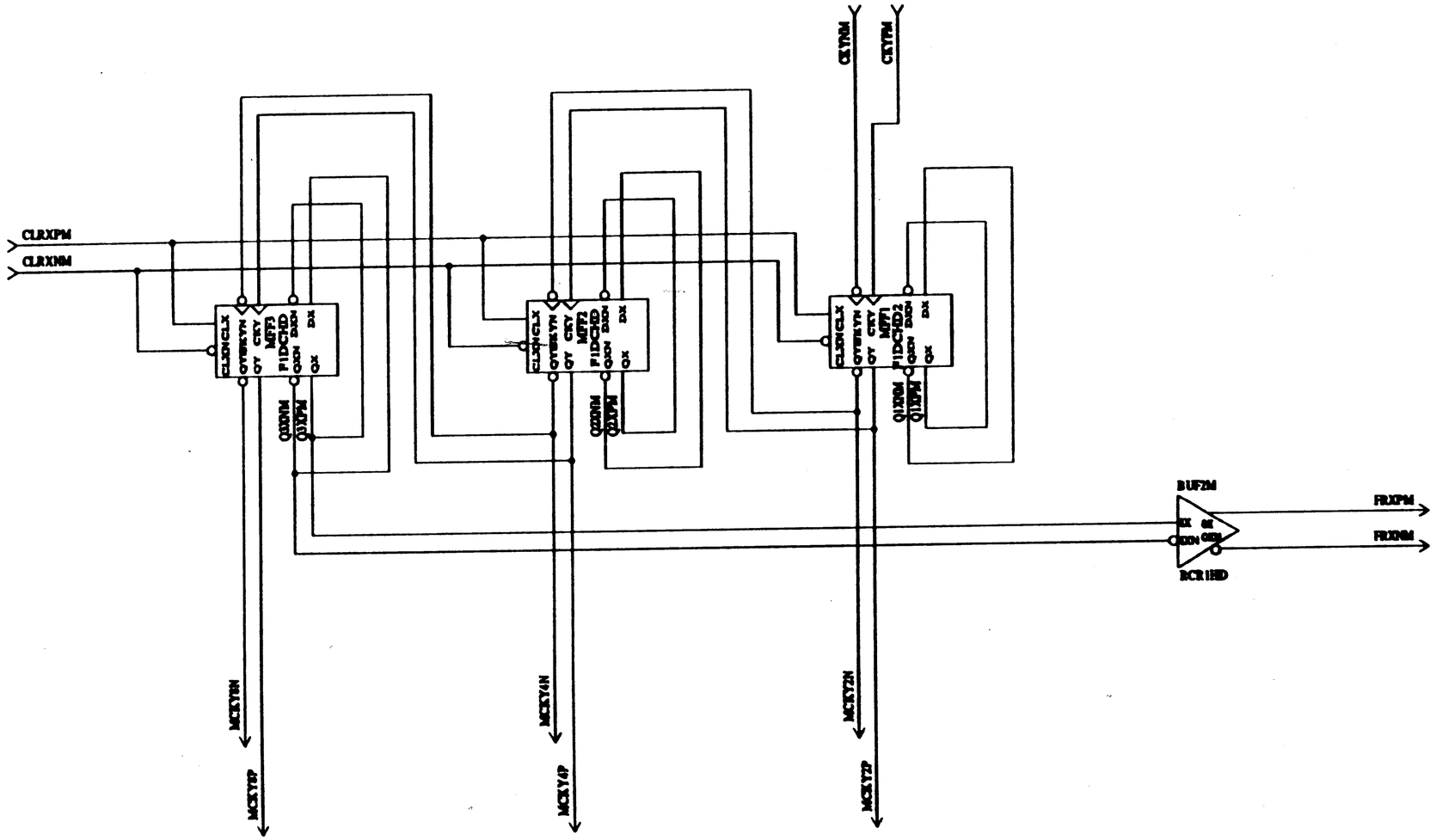


FIGURE #28. 3 BIT RIPPLE COUNTER FOR THE MULTIPLEXER

Likewise, the CKY4PM clock signal will be a 625MHz clock signal which is half of the CKY2PM clock frequency, and finally the CKY8PM clock signal will be 312MHZ. An important point to be remembered however, is the gate delay worth of time between the change of logic states of each of the clocks. The beauty behind this gate delay is better understood upon analyzing the combination of the 8 input data channels to 4 data channels by the first stage of (F4DAHD) mux latches ML1-ML4.

To begin, a single delay line is used to control the input high speed clock in order to align the MCKY8P and MCKY4P clock signals with the eight 300MHz data channels input on MDATA0-MDATA7 to the first stage of mux latches ML1-ML4. This stage then carefully combines these eight channels of data into the 4 channels MUX04P, MUX26P, MUX15P, and MUX37P. Focusing on ML1, its select leads are controlled by the 312MHz clock signal MCKY8P, while the data enable is controlled by the next higher frequency 625MHz clock signal MCKY4P. When MCKY8P is in a logic low state, the mux latch will acquire data from input MDATA0, and when it is in a logic one state, the mux latch will acquire data from the other input MDATA4. The 625MHz clock MCKY4P will control the latching of the data from the input data channels when it is in a logic high state, and will enable data through the latch transparently to MUX04P when it is in a logic low state. This is the beauty of the gate delay of the ripple counter as shown in the timing diagram in figure #29. The 312MHz clock will always change state to select new data a gate delay after the 625MHz clock has latched the present data.

This method is repeated in the second stage of mux latches ML5 and ML6 where the 4 data channels consisting of the previous 8 bits of data are combined into the 2 data channels MEVENP and MODDP. The selecting of the 4 channels of data is performed by the 625MHz clock MCKY4P, which previously controlled the latching and enabling of the data through the first stage of mux latches. The latching and enabling of the data in the mux latches ML5 and ML6 is controlled by the 1.25GHz clock MCKY2P. Again, as in the first stage, the select clock MCKY4P will not change state to select new data until a gate delay after the MCKY2P clock has latched the present data being selected. The third and final stage of multiplexing consists of the (F2DAHD) mux flip-flop MFF4, which takes the 2 data channels MEVENP and MODDP and interleaves a bit from each channel creating a serial data bit stream. Note, that the selecting between the two channels is performed by the 1.25GHz clock MCKY2P which

previously controlled the latching and enabling of the previous stage. The clocking of the data into the mux flip-flop is performed by the 2.5GHz clock which also controls the 3 bit ripple counter. Since the flip-flop is positive edge triggered, with every positive transition of the CKPM 2.5GHz clock, the flip-flop MFF4 will latch and hold the data bit which is being selected by the 1.25GHz clock MCKY2P. Immediately following this transition, the MCKY2P clock will begin selecting the next data

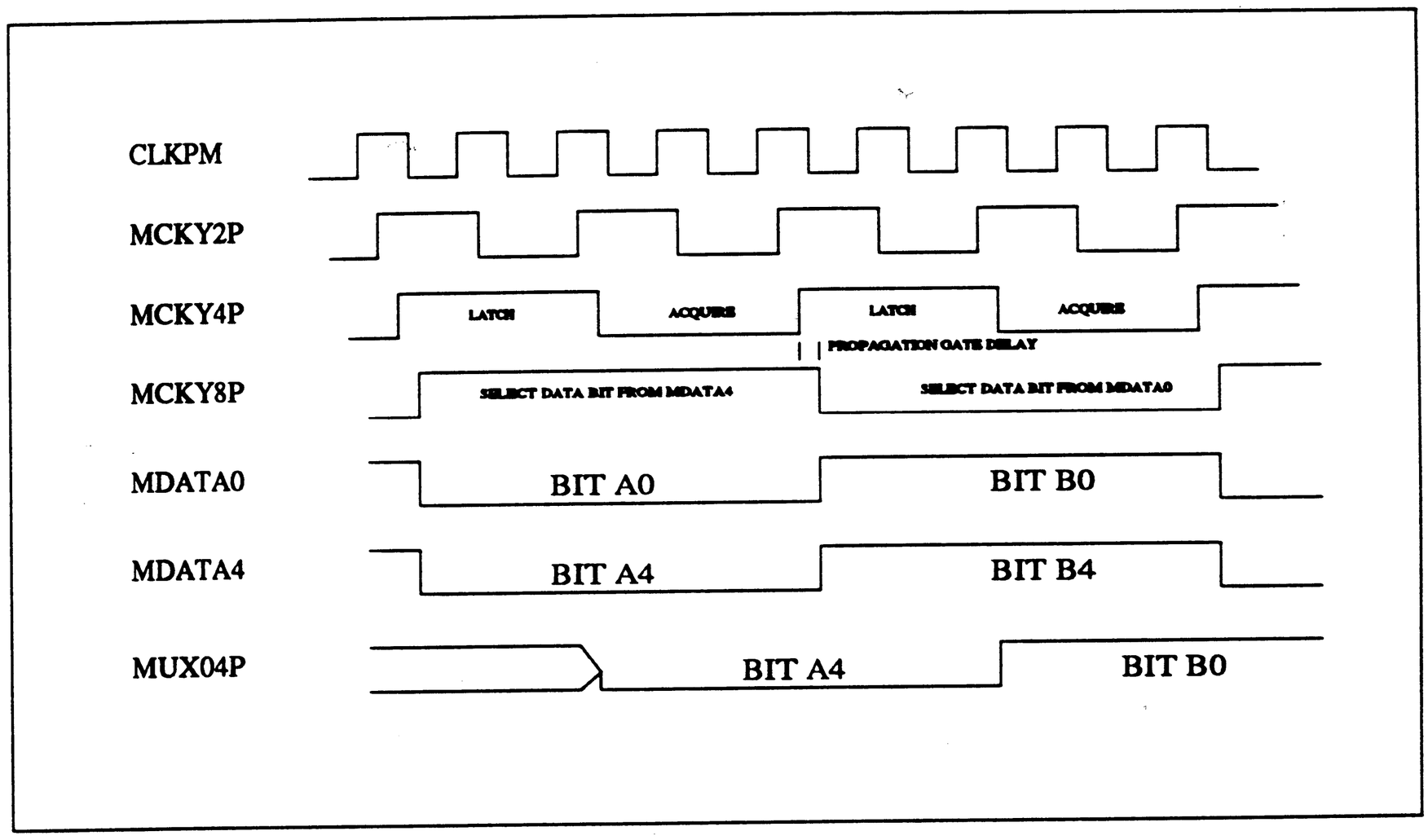


Figure #29. First Stage of Multiplexing Operation

bit to be latched. The mux flip-flop is required in this final stage over a mux latch to protect against edge jitter and meet the requirements listed previously in the system specifications. If a mux latch were used, the duty cycle on the 2.5GHz clock CKPM would have to be accurately controlled 50% duty cycle to maintain equivalent widths of each output bit. By using a flip-flop, only the control of a periodic rising edge of the 2.5GHz clock CKPM is required, therefore yielding better control of the output data bit widths and minimization of any edge

jitter.

The differential frame clock FRPM and FRNM serves two purposes. First, the positive transitions of FRPM signify the beginning of a new byte to be transmitted, producing what is known as a framing pulse around the transmitted data. Secondly, this clock can be used external to the circuit to control the delivery of new byte of data to the Time Division Multiplexer by clocking an external register that holds the 8 channels of data to be multiplexed. The timing for this is perfect since the first stage of mux latches ML1-ML4 will be ready to accept a new word of data well before the frame clock propagates off chip to clock the register which will supply the new data word. The differential transmit clock TRPM and TRNM serves only as a convenience to the system designer in order that a second 2.5GHz crystal can be eliminated for the Time Division Demultiplexer by making use of this transmit clock to drive the CKPDM and CKNDM clock inputs of that circuit. Since this clock is intimately related to the transmitted data, it is conceivable that upon transmitting this clock followed by a clock recovery scheme, this signal could then drive the Time Division Demultiplexer, and possibly eliminate the need for the shifting of data in time and space. However, as will be seen in the next chapter which describes the operation of the Time Division Demultiplexer, the shifting provision in conjunction with an extra 2.5GHz crystal is a much more practical solution over the added expense of using a second transmission line for the transmit clock.

As will be seen later in chapter 8, the layout of this circuit requires the use of three separate Band Gap reference circuits each of which consumes 38mW of power. Upon including these with the rest of the circuit, a typical power consumption of 0.667W results. In comparison to an equivalent NTT 8:1 channel Time Division Multiplexer,[1] this design uses 6 fewer slave latches resulting in a power savings of 36mW, which is 5% of the total power consumption of the multiplexer circuit. In retrospect, this circuit meets all of the specifications and system requirements listed previously and latter simulations in chapter 9 will demonstrate typical circuit operation at 3.2Gbit/s.

7. 1 Channel to 8 Channel Time Division Demultiplexer

The 1 channel to 8 channel Time Division Demultiplexer separates a 2.5Gbit/s single continuous serial stream of data into eight 300MHz data channels. In total, the circuit has 8 inputs and 10 outputs as shown in figure #30. As with the Time Division Multiplexer, two of the inputs, CLKPDM and CLKNDM, are used to apply a 2.5GHz ECL level differential sinusoidal input clock to the circuit to clock a 3 bit ripple counter. A clear input, CLRDM, is also necessary for testing purposes to clear the counter into a known count state. Three of the last five inputs, SHIFT1, SHIFT2, and SHIFT4, control the count sequence of the 3 bit ripple counter by either enabling or inhibiting the toggling of each stage. Finally, the last two inputs, DATAP and DATAN, are used to supply the 2.5Gbit/s ECL level differential data bit stream to the circuit. Of the 10 outputs, eight of them represent the eight 300MHz demultiplexed data channels, DATA0-DATA7. They supply bytes of data framed by the ECL level differential outputs, FRPDM and FRNDM, which are a divide by 8 version of the input 2.5GHz differential clock.

As with the multiplexer, only the positive sense of the differential signals will be discussed since the signals are defined to be complementary. We will begin the analysis of the circuit with the 3 bit ripple counter shown in figure #31. This counter differs slightly from the counter described for the multiplexer in the previous chapter, in that it has a second mode of operation besides the ripple count mode. When the external circuit inputs SHIFT1-SHIFT4 are in a logic low state, the S0Y inputs to the three (F2DCHD) mux flip-flops DMFF1-DMFF3 are biased to a logic high state which selects the D1X data. The D1X input of each flip-flop is connected to its own QXN output placing the counter into a toggle mode of operation as was previously described for the Time Division Multiplexer. Therefore, when the 2.5GHz clock input CLKPDM clocks the flip-flop DMFF1, the output DMCLK2P will be a 1.25GHz clock signal which will then drive DMFF2. In turn, the DMFF2 output will be a 625MHz clock signal, DMCLK4P, which will then drive the flip-flop DMFF3. Finally, DMCLK8P will be a 312MHz clock signal generated by flip-flop DMFF3 which becomes the frame clock FRPDM, and controls the final output of data. The second mode of operation is the toggle inhibit mode which occurs when either or all of the external circuit inputs SHIFT1-SHIFT4 are in a logic high state. This results in the S0Y inputs to the three mux flip-flops DMFF1-DMFF3 to go into a logic low state which selects the D0X data. Since, the D0X input is connected to the

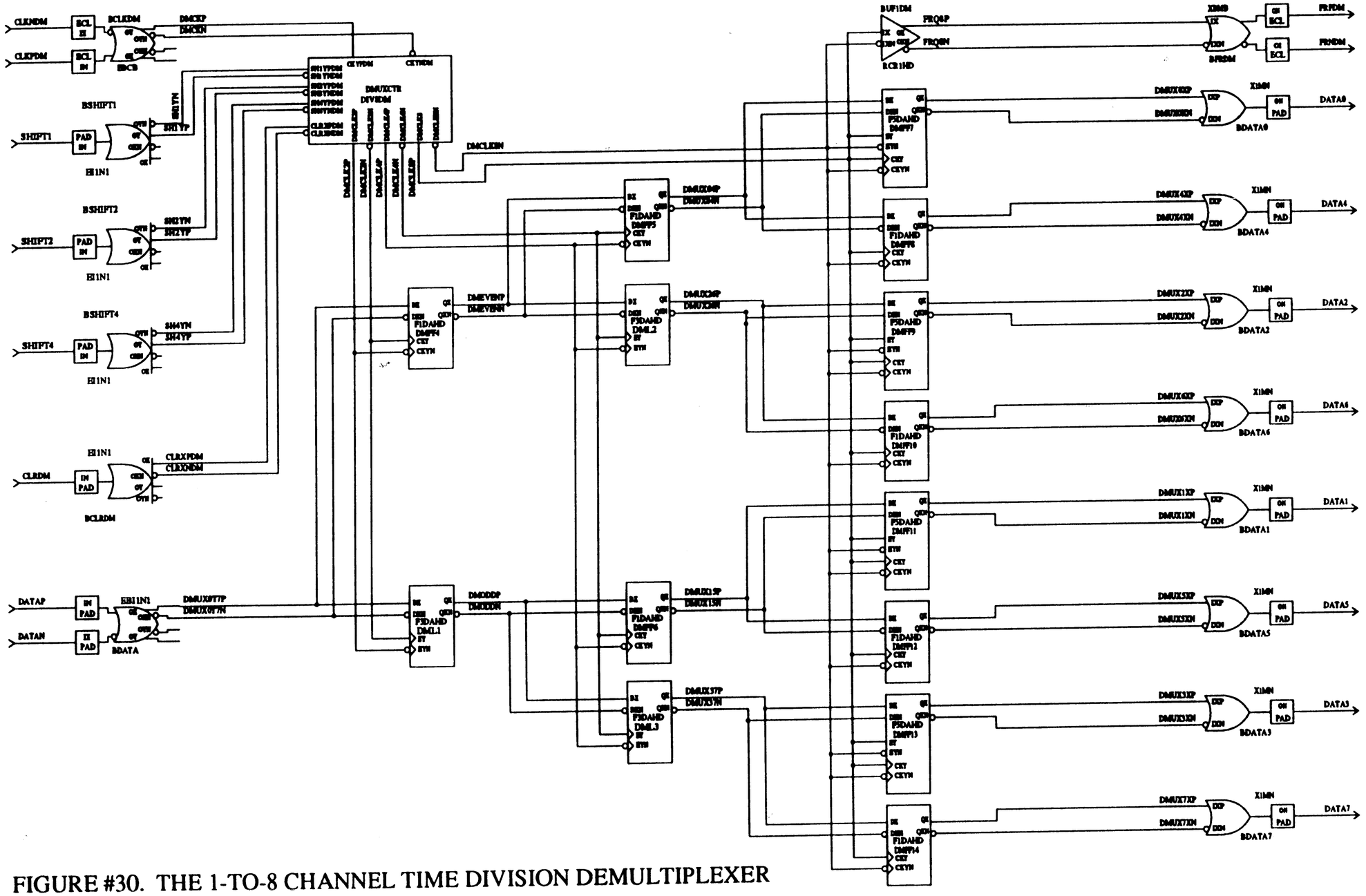


FIGURE #30. THE 1-TO-8 CHANNEL TIME DIVISION DEMULTIPLEXER

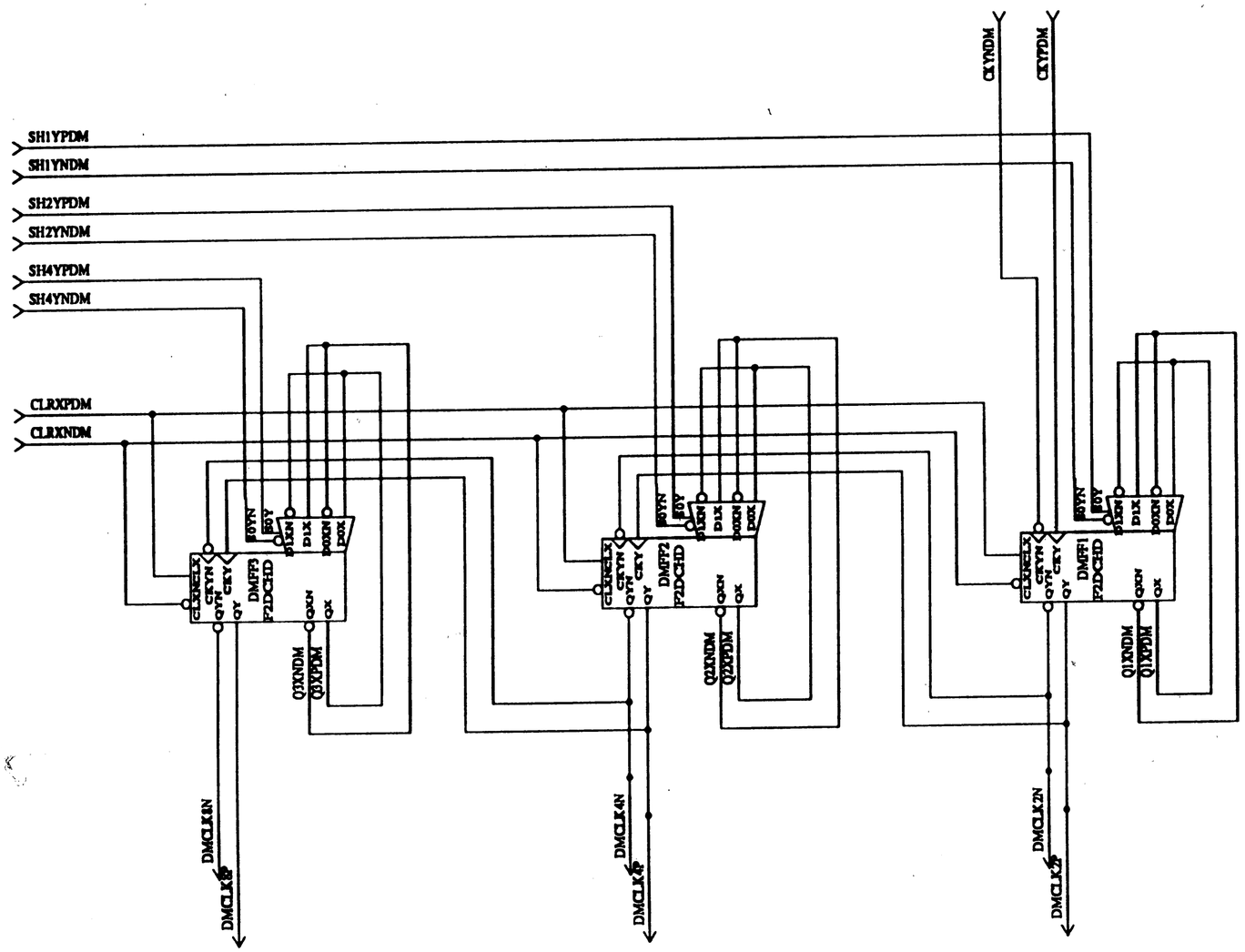


FIGURE #31. THE 3 BIT RIPPLE COUNTER FOR THE DEMULTIPLEXER

QX output of the same flip-flop, then with every positive triggering clock edge, the flip-flop will re-read the old data and the toggle will be inhibited. Each of the three flip-flops can be controlled independently, with SHIFT1 controlling DMFF1, SHIFT2 controlling DMFF2, and SHIFT4 controlling DMFF3. This is necessary in order that shifts in time and space of the input data can be performed in one, two, or four bit increments to realign the data with the proper output data channel within the proper byte of information. For example, a logic high at the SHIFT1 input just prior to a positive active clock edge on CLKPDM will disable the toggling of flip-flop DMFF1. This will result in the redirection of the data headed for the DATA0 output to to to the DATA1 output. Likewise, DATA1's data will go to DATA2,...,and DATA7's data will go to DATA0. This not only describes a shift in time of one bit of information for outputs DATA0-DATA6, but also describes a shift in space where data bit8 of one byte of information becomes bit0 of the next byte of information. Note, that this feature is very helpful to the systems designer for it allows for not only single bit shifts in time and space, but any increment between 1-7 bits by using combinations of the external shift inputs. Now that the examination of the two modes of operation for the 3 bit counter is completed, the remainder of the Demultiplexer's logic can be examined along with how the ripple counter interfaces with that logic to control the separation and data channel alignment of the input serial stream of data.

To begin, the 2.5Gbit/s differential input data enters the circuit at the inputs DATAP and DATAN. A single delay line is necessary for either the clock or data inputs prior to the chip to align the changing edges of the data with the inactive falling edges of the clock CKPDM as shown in figure #32. This data is then amplified by the differential input buffer BDATA and is directed to the inputs of DMFF4 (F1DAHD) and DML1(F3DAHD). These two function blocks are controlled in parallel by the 1.25GHz clock signal DMCLK2P as shown in the circuit figure #30. Since DMCLK2P clocks the CKYN input of the flip-flop DMFF4, then with every negative falling edge of the clock DMCLK2P the data bit on signal DMUX0T7P will be transferred through to DMEVENP. The data latch DML1 operates differently. Since DMCLK2P controls its EYN input, with every logic low state on DMCLK2P data bits from DMUX0T7P will be allowed to move transparently through the data latch to signal DMODDP. Furthermore, a logic high state on DMCLK2P will latch whatever data bit is located within the data latch, and will inhibit the flow of any new data from signal DMUX0T7P to signal

DMODDP.

The following waveforms in figure #32 describe this first stage. A close examination shows

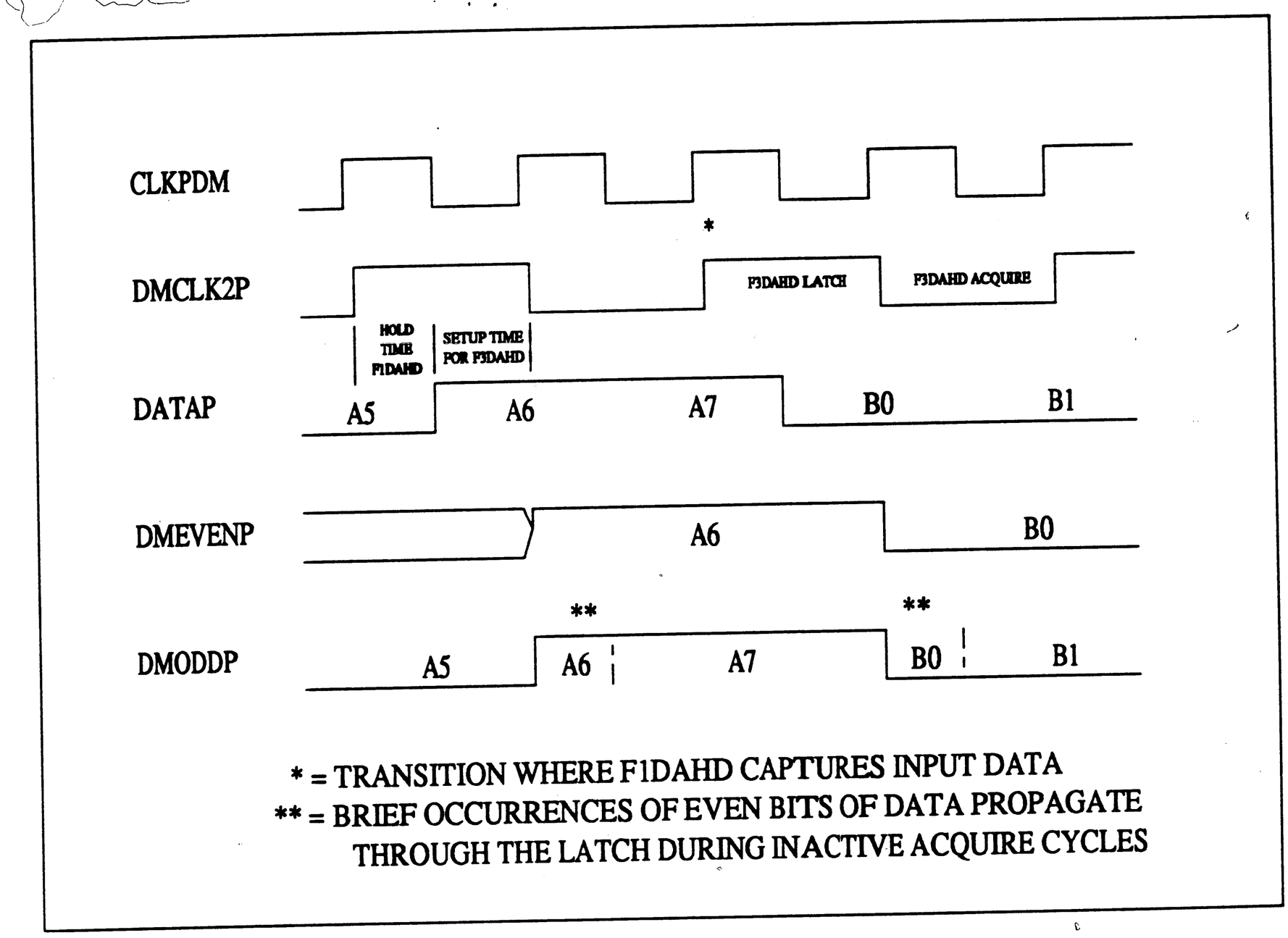


Figure #32. First stage of the Demultiplexer

the DMFF4 flip-flop separating out every other even bit of data and channeling the information onto the signal DMEVENP, while the data latch DML1 separates out every other odd bit of data and channels the information onto the signal DMODDP. This also reiterates the need for the exterior delay line which controls the amount of setup time for the odd data bits prior to the data latch DML1. If the delay is aligned such that changes of data occur just after the rising edge of the input clock CKPDM, then setup time will be maximized for the odd bits of

data to DML1. However, the maximization of this setup time minimizes the hold time for the even bits of data being captured by the flip-flop DMFF4. This explains the logic behind aligning the data changes to the falling clock edges of CKPDM to allow for equal setup and hold times for the even and odd bits of data.

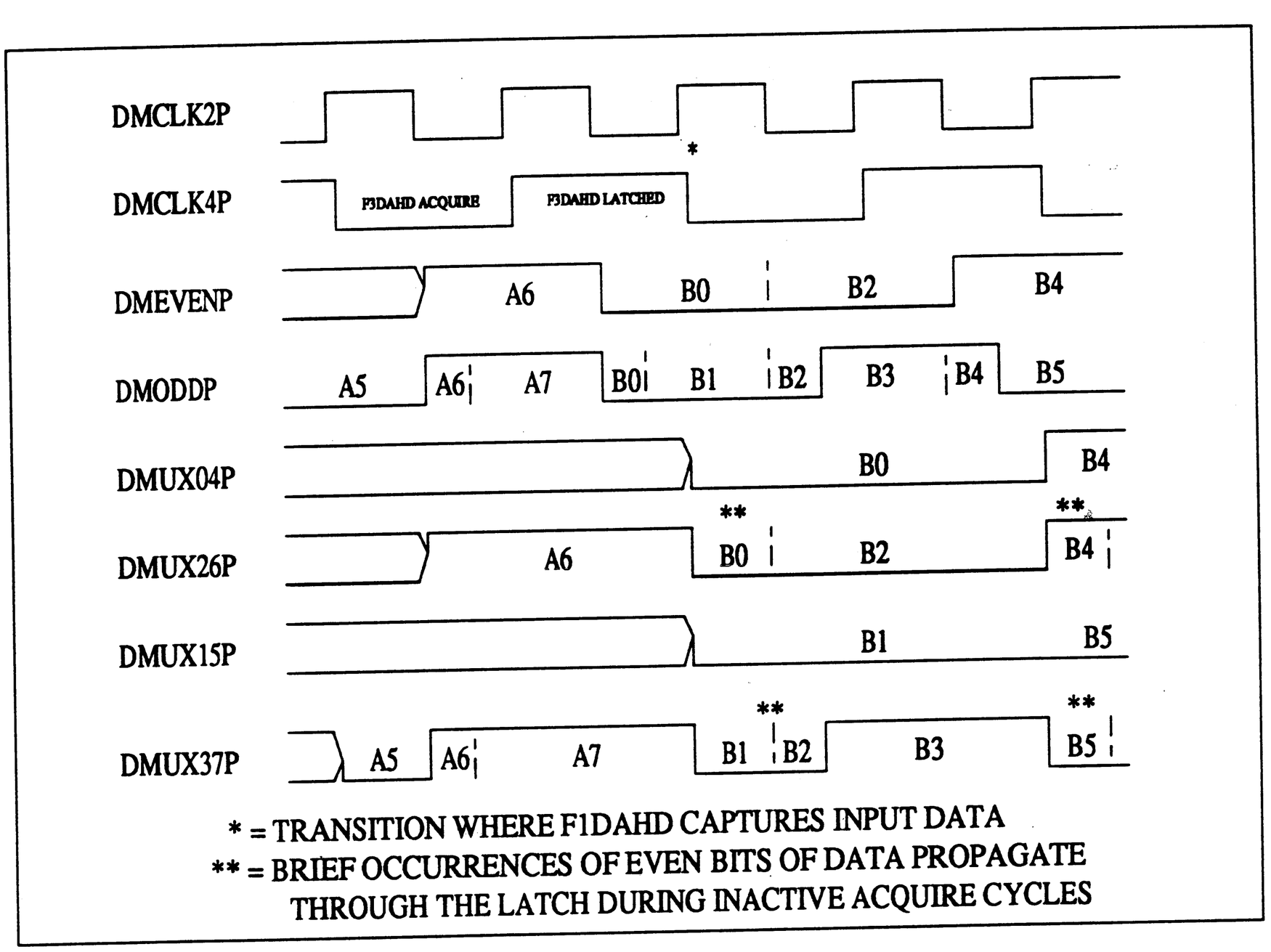


Figure #33. Second Stage of Demultiplexer

After the input data has been separated onto signals DMEVENP and DMODDP, they are directed to the second demultiplexing stage consisting of two parallel first stages all of which are controlled by the 625MHz clock DMCLK4P shown in figure #30. Again, let's analyze the following waveforms of figure #33 which best describe the logic. These waveforms show that

the even data bits on signal DMEVENP are separated by the (F1DAHD) flip-flop DMFF5 and (F3DAHD) data latch DML2. The bits 0 and 4 get channeled to the signal DMUX04P on negative clock edges of DMCLK4P, while bits 2 and 6 get channeled to the signal DMUX26P

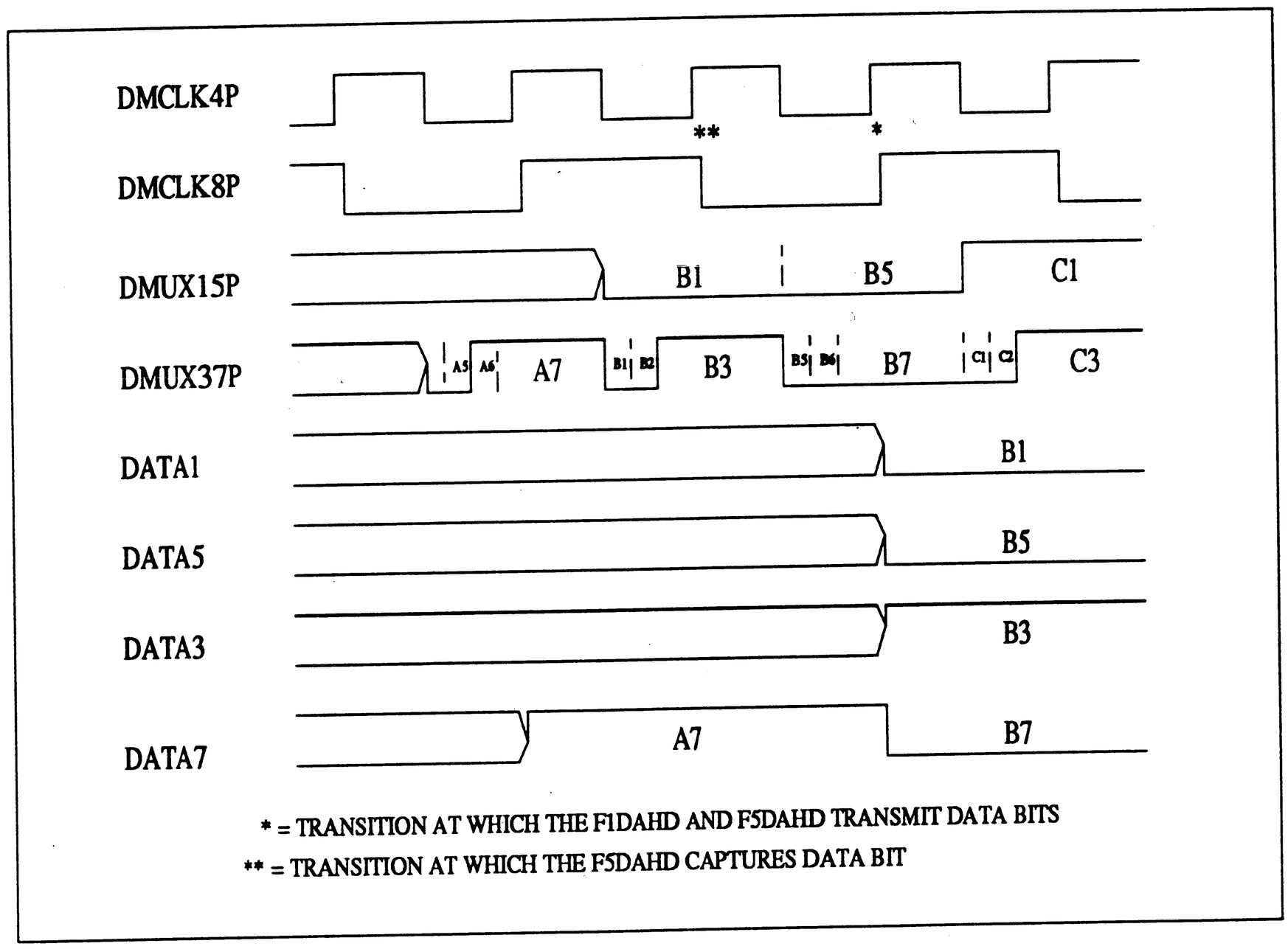


Figure #34. Third Stage of Demultiplexer

with logic low levels on DMCLK4P. Likewise, the data bits on signal DMODDP are separated by the (F1DAHD) flip-flop DMFF6 and the (F3DAHD) data latch DML3, and the bits 1 and 5 get channeled to the signal DMUX15P on negative clock edges of DMCLK4P, while bits 3 and 7 get channeled to the signal DMUX37P with logic low levels on DMCLK4P.

The third and final demultiplexing stage consists of 4 parallel first stages all controlled by the 312MHz clock DMCLK8P as shown in the figure #30. However, in this final stage the data latch is replaced with a Three Stage flip-flop (F5DAHD). This flip-flop is necessary for the realignment of data bits 0-4 such that all 8 bits of data will arrive at the outputs concurrently after positive clock edges of DMCLK8P. For simplicity, since the even channel outputs operate identically to the odd channel outputs, only the signals DMUX15P and DMUX37P will be explained in the following waveforms shown in figure #34. Note, that the (F5DAHD) flip-flops DMFF11 and DMFF13 capture and hold the data bits located on the signals DMUX15P and DMUX37P respectively with every negative clock edge of DMCLK8P. These data bits are stored in the first stage of the flip-flop until the following rising clock edge of DMCLK8P, during which time the data bits are transmitted to the second and third stages of the flip-flop and finally to outputs DATA1 and DATA3 respectively. On the other hand, the (F1DAHD) flip-flops DMFF12 and DMFF14 only capture data bits from signals DMUX15P and DMUX37P with every positive rising clock edge of DMCLK8P, therefore they effectively ignore the previous data bit which is captured by the DMFF11 and DMFF13 flip-flops. Furthermore, instead of holding the data bit until the next clock edge of DMCLK8P, these flip-flops transmit the data bits immediately to outputs DATA5 and DATA7 respectively. Since all of the flip-flops transmit data bits to the outputs on positive clock edges of DMCLK8P, along with the fact that the layout of the slave latches are identical for both the F5DAHD and F1DAHD flip-flops, results in the 8 bits of data arriving at the outputs in parallel which minimal skew across all outputs. The DMCLK8P clock is also buffered through the differential ECL inverter (RCR1HD) BUF1DM and the output buffer BFRDM where it becomes the frame clock FRPDM and FRNDM. The differential inverter is necessary to match the delay that the data bits experiences traversing the slave latch of the flip-flops, so as to minimize the skew between the frame clock and the output data bits.

As will be seen in the next chapter, the layout of this circuit requires the use of four separate Band Gap reference circuits each of which consumes 38mW of power as previously described for the Time Division Multiplexer. Upon including this power with the rest of the circuit, a typical power consumption of 0.77W results. In comparison to an equivalent NTT 1:8 channel Time Division Demultiplexer,[1] this design uses 12 fewer slave latches resulting in a power savings of 72mW, which is a 8% savings of the total power consumption for the demultiplexer

circuit. In retrospect, this circuit meets all of the specifications and system requirements listed previously and latter simulations in chapter 9 will demonstrate typical circuit operation at 3.1Gbit/s.

8. XY Mask Layout of the Integrated Circuit

A cell based semi-custom approach was used in the layout of the integrated circuit which consists of both the Bit Interleaving Time Division Multiplexer and Time Division Demultiplexer. This type of an approach leaves the designer with the most flexibility in placement and routing of the completed internal building blocks since the internal standard cell diffusion set is used as a common template for each of the internal building blocks to be interconnected upon. Therefore, with a common logic template, the designer is free to place any of the internal building blocks anywhere an unused standard cell template exists similar to the Motorola MCA3 series third generation ECL Macrocell gate array approach which consists of an internal cell of 76 transistors and 76 resistors which is divided into quadrants.[11] However, in this gate array, the diffusion is fixed and rows of logic cells cannot be moved to accommodate the routing interconnect in the channel. In this semi-custom approach, the logic rows consisting of the internal cell can be moved because the diffusion set is not fixed. The drawback to this approach is the potential of increased parasitic routing resistances and capacitances within a building block layout because of the choice of the number of transistors and resistors within the standard cell and the placement of these elements to form the template. However, if the standard cell is constructed wisely this drawback can be minimized.

The standard cell, shown in figure #35, for this work consists of 22 transistors, 6 2kΩ resistors, 12 1.51kΩ resistors, and 4 600Ω resistors. This selection of components was done in order that the use of a single template for the layout of the F3DAHD data latch and a double template for the layout of the F1DCHD positive edge triggered D-type flip-flop would produce the most optimum layouts for these building blocks. This happens to provide very optimized layouts for the other internal building blocks as well, since the data latch is the underlying circuit in all but one of the internal building blocks as was found in chapter 5. The RCR1HD differential inverter has the worst of the layouts, but it is only used in both circuits as a low frequency buffer. A similar concept was used with the buffer building blocks, where a common buffer standard cell diffusion template, shown in figure #36, consisting of 22 various sized transistors, 2 schottky diodes, and 26 various sized resistors was developed. This buffer standard cell can accommodate any of the 5 buffer building blocks, by also making use of single or double placements of this template. The buffers which require double placements of this buffer standard cell happen to have differential inputs and or outputs, therefore, the

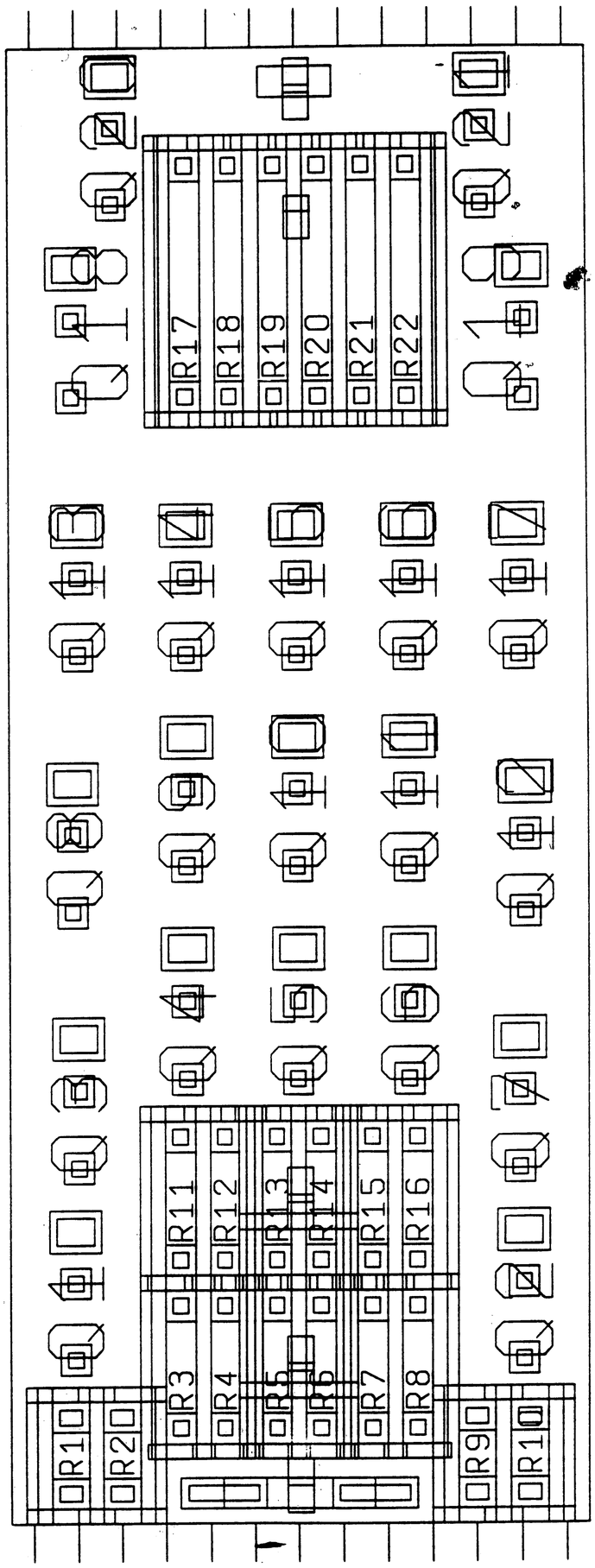


Figure 35. Layout of Internal Standard Cell

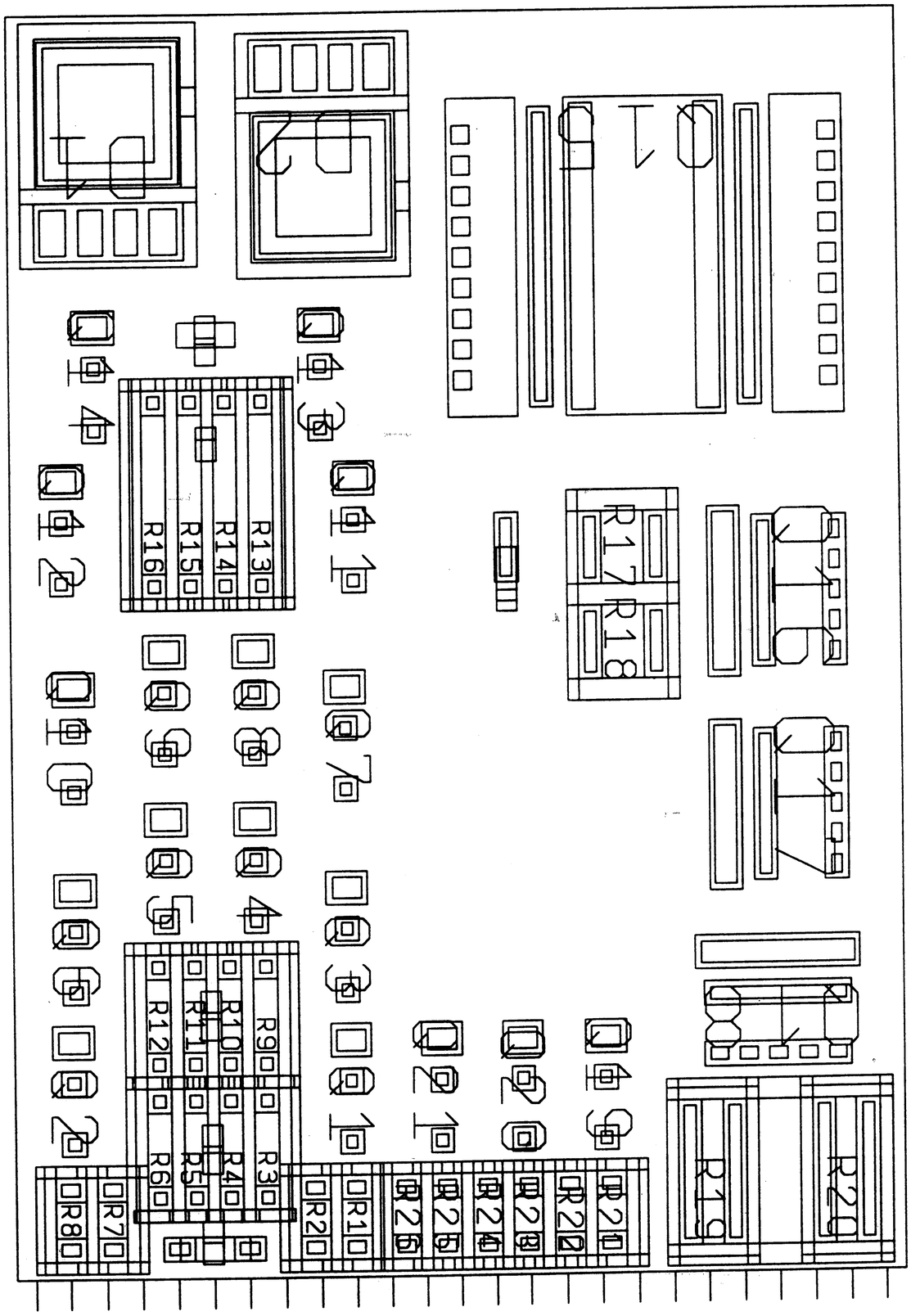


Figure 36. Layout of External Buffer Cell

standard buffer cell is constructed such that side-by-side placements result in an overall symmetric buffer.

Once these two standard cells are developed, a diffusion set for the overall gate array is put together as shown in figure #37. The final chip size is approximately 120 mils by 120 mils and is limited by the bond pad to bond pad spacing as well as the manufacturable wirebond length from the die to the TriQuint ceramic package.[12] Because of this pad and wirebond limitation, the internal area of the chip becomes larger than it need be. Despite this, the area is filled with internal standard cells even though it is known that many of them will be unused. In total, 40 buffer standard cells are placed around the periphery of the die to handle the 40 possible I/O pins of the package, along with 4 Band Gap regulators, located in the middle of each side of the integrated circuit. Internally, 6 identical logic rows are constructed each consisting of 20 internal standard cell templates with a supporting Band Gap regulator in the middle of the row. Once the die diffusion set is fixed, the individual Multiplexer and Demultiplexer layouts are started by placing the defined building blocks onto a unused standard cell site with the results shown in figure #38.

The Time Division Multiplexer is placed and routed on the top half of the integrated circuit, leaving the bottom half of the die for the Time Division Demultiplexer. The Multiplexer only uses two of its three allotted logic rows, while the Demultiplexer uses all three of its allotted rows. Two critical timing issues were addressed during layout of the Multiplexer circuit. First, from the logic discussion in chapter 6, it is apparent that the high speed clock path be made as short as possible to the first stage of the ripple counter MFF1, and the output Mux-type flip-flop MFF4, as well as the output data path from MFF4 to the output buffer BOUT. Therefore, these two internal building blocks are placed as close as possible to the clock buffer BCLKM and the output buffer BOUT at the top center of the die. Furthermore, since the most critical high speed timing relationship on the circuit exists between MFF1 and MFF4, both circuits are controlled by a common Band Gap Regulator. The second timing issue results from the system specification that only one delay line be used to align the high speed clock to the 8 data channels. This places a requirement on the circuit that all 8 data paths to the first stage of the multiplexer have equivalent setup and hold times. A large part of this time is manageable by choosing identical buffers and controlling all of them with the same Band Gap Regulator, as is done in the layout. The remainder of the time depends on the various routing

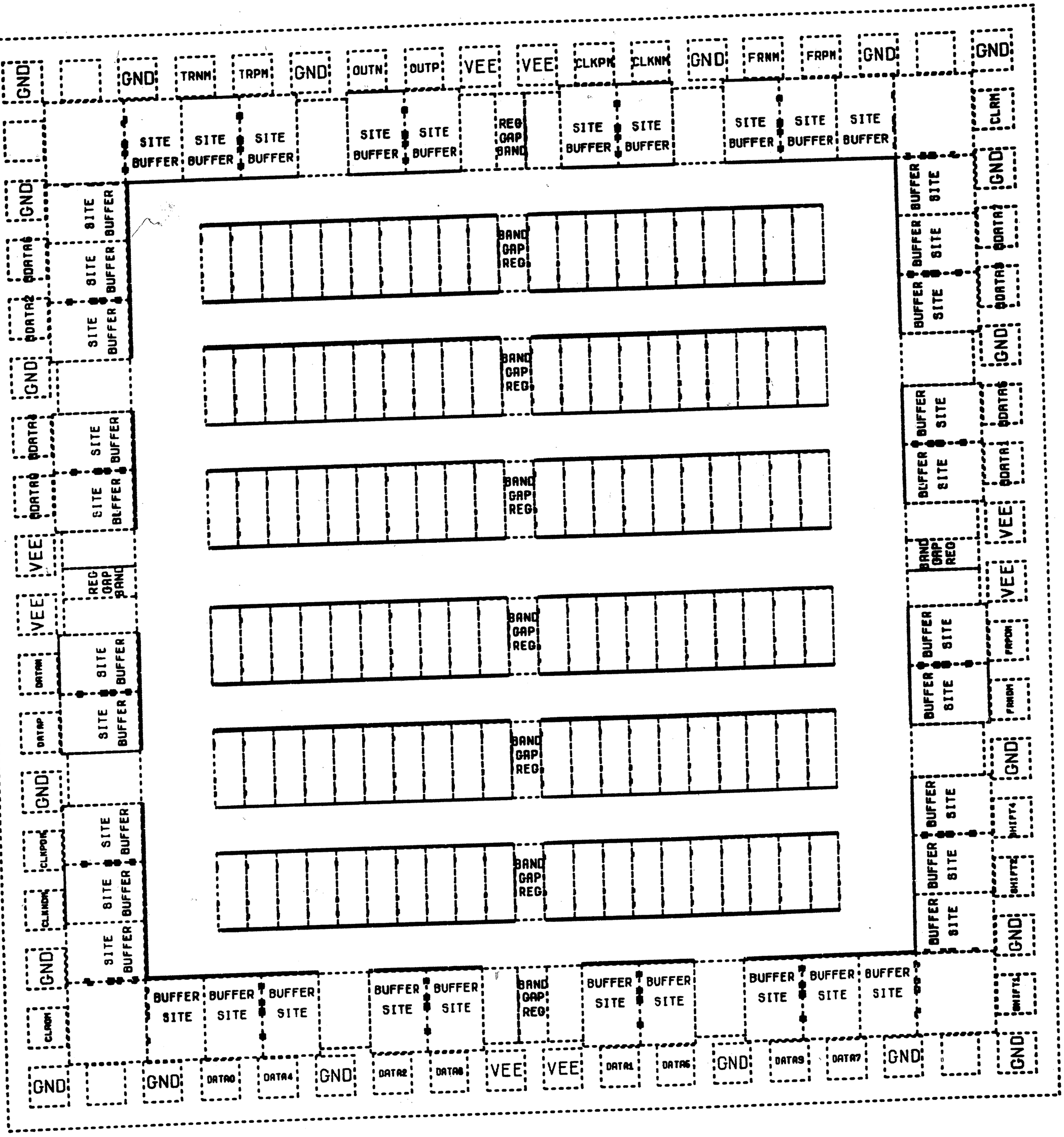


Figure 37. Layout of Generic Diffusion Set for Die

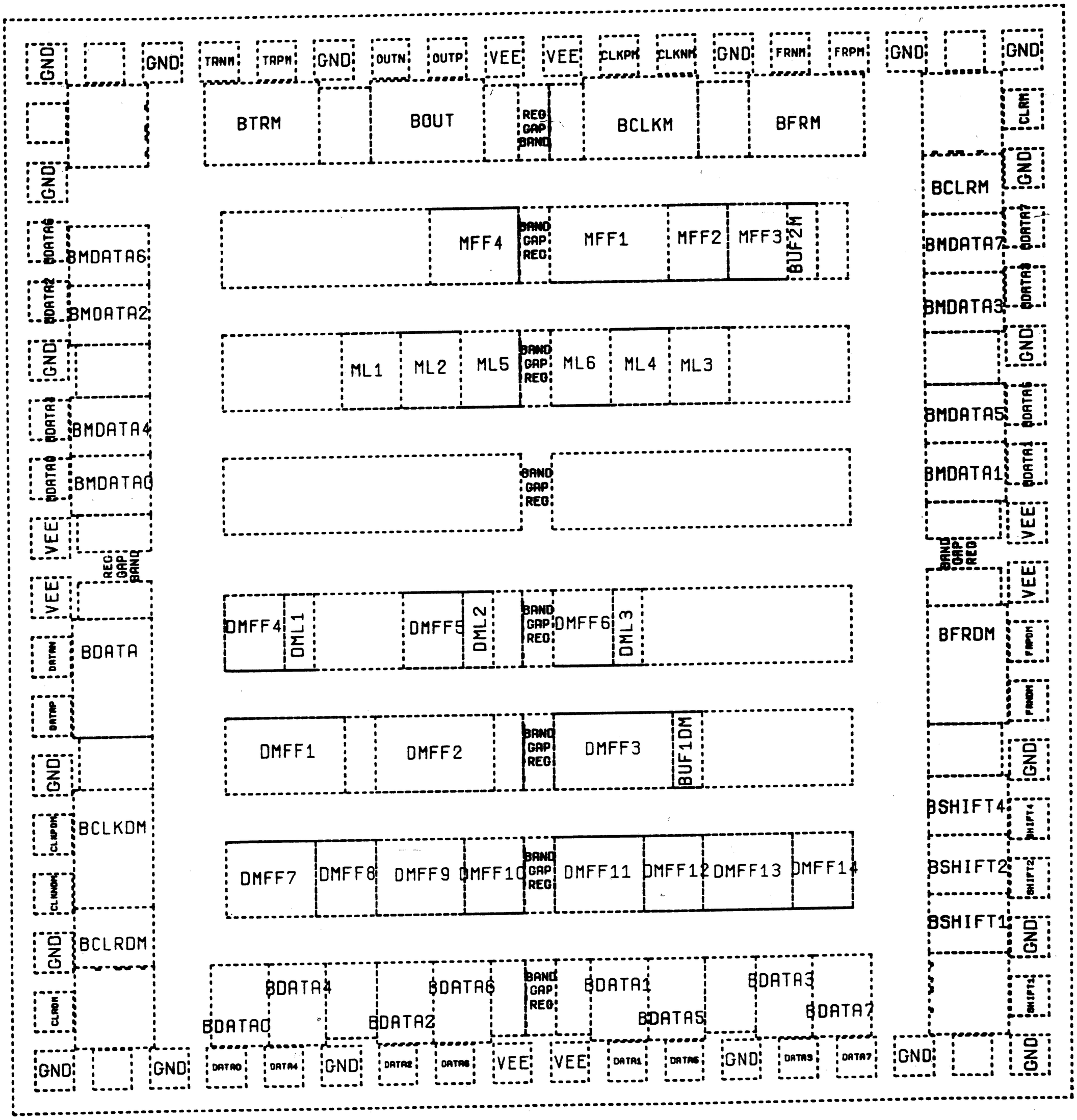


Figure 38. Placement of Building Blocks for Die

capacitances of the input data channels after they have propagated through the input buffers. Therefore, by selectively routing each of these signals, making use of the two different metal level capacitances, all of the signal capacitances are matched to 0.5 picofarad (pF) accuracy. Two critical timing issues were addressed with the layout of the Demultiplexer as well. As with the Multiplexer, the input high speed clock path and data path are critical and should be shortened. Therefore, the first stage of the ripple counter DMFF1 and the first stage of the demultiplexer are placed as close as possible to the BDATA and BCLKDM differential input buffers. Furthermore, these building blocks are placed as close as possible to each other in order to maximize the performance between the counter and this demultiplexing stage. One might question why these building blocks are not placed similar to the approach taken with the Multiplexer in the top half of the chip. This can't be done as a result of the second critical timing issue of the less than 100psec skew between the 8 output data channels. The only way to minimize the skew, is to place all of the flip- flops in the last stage of the demultiplexer as close as possible to their respective output buffer and control them with the same Band Gap Regulator. Furthermore, all of the 8 output buffers must be in close proximity and need to be controlled on the same Band Gap Regulator as well. The only place on the integrated circuit where all of these conditions can be met is in the bottom logic row as shown in figure #34. As with the multiplexer, routing capacitance matching is done between the last stage flip-flops and their output buffers to further minimize the possible skew, resulting in a 0.1pF matching of capacitances on these signals.

9. Circuit Simulation Results

The connectivity of both circuits along with the parasitic routing capacitances were extracted from the layout of the integrated circuit and were simulated using a SPICE like simulation tool called ADVICE. Due to constraints on the length of the possible simulation, with the Time Division Multiplexer, DC ECL low levels of -1.6 volts were applied to the mdata0, mdata3, mdata5, and mdata7 inputs, while DC ECL high levels of -0.8 volts were applied to the other four inputs. These input conditions were chosen in order that the pseudo random waveform 0110010101100101O be applied to the final and most critical stage of the Multiplexer (i.e MFF4) to verify its functionality and frequency performance over the spectrum of possible signal frequencies that this flip-flop might experience. Ideally, pseudo random waveforms should be input to the entire circuit to verify the functionality and frequency performance of all of the stages of the Multiplexer. However, this would have resulted in lengthy and expensive simulations, that most likely would not have produced any new information. On the other hand, simulations for the Time Division Demultiplexer were done the following input pseudo random waveform 01011001101001001111l which ideally tests this circuit over the spectrum of possible signal frequencies.

Both circuits were simulated at 2.5Gbit/s data rates over the following typical, worst case fast, and worst case slow conditions, shown in Table 1.

	Worst Case Fast	Typical	Worst Case Slow
VEE Power Supply	-5.7 volts	-5.2 volts	-4.7 volts
Transistor Gain	200	125	50
Sheet Resistance	-20%	nominal	+20%
Junction Capacitances	-5%	nominal	+5%
Ambient Temperature	100 degrees C	25 degrees C	0 degrees C

TABLE 1. Simulation Conditions

Typical simulations were also extended to higher data rates to determine the point at which

both circuits stop functioning. The results of these simulations follow.

The simulation results for the Time Division Multiplexer are shown in figures #39 - #45. First, figure #39 demonstrates a typical circuit functionality with an output pseudo random data rate of 2.5Gbit/s. Note, that the first bit of the transmitted byte begins at 4.5 nanoseconds, and from this point on the positive sense of the data stream is 01101010011. The positive frame clock edge which signifies the beginning of a new byte of information is delayed 500 picoseconds by the delay through the three stages of the ripple counter, the differential RCR1HD inverter, and the output differential buffer. Depending on the system designers application, this delay could be tolerable. Otherwise, decoding of the ripple counter could be done in order to produce a rising frame clock to meet any system design. The simulation output for the transmit clock shows a 2.5GHz sinusoidal waveform 600mV p-p. This degraded signal swing is a sign that the output buffer has run out of bandwidth, and needs to be redesigned to handle higher frequencies if the system designer choose to make use of this clock. Figure #40 is an exploded view of the output data stream demonstrating a controlled data crossover point of -1.25 volts which is exactly 50% of the signal amplitude. This figure also shows a very controlled peak to peak signal height well within the +/- 200mV specification, as well as 150 picosecond rise times and 100 picosecond fall times. The single 400 picosecond pulse, resulted in deviations of the zero crossing intervals from the fastest pulse to the slowest pulse of 28ps which represents 7% worth of edge jitter. The 800 picosecond pulse zero crossing intervals are much better controlled with a worst case deviation of 23ps which represents 3% edge jitter. Figure #41 represents the functionality of the Time Division Multiplexer under worst case fast conditions. Note, that the same data stream is produced, and the orientation of the rising frame clock is much better since the internal delays are faster. Likewise, the higher simulated gains in this condition improves the bandwidth of the transmit buffer producing a solid 2.5GHz 800mV p-p sinusoidal waveform. Figure #41 gives a exploded view of the output data stream, which shows the cross over point has moved to -1.19 volts which is still at 50% of the over all signal amplitude. This figure also demonstrates good control of pulse width and amplitude variations for minimal signal jitter. Figures #43 & #44 demonstrate the same results for the circuit under worst case slow conditions with the only concern being the slower delay of the rising edge of the frame clock as a result of the slower internal building block delays. The last figure #45, demonstrates

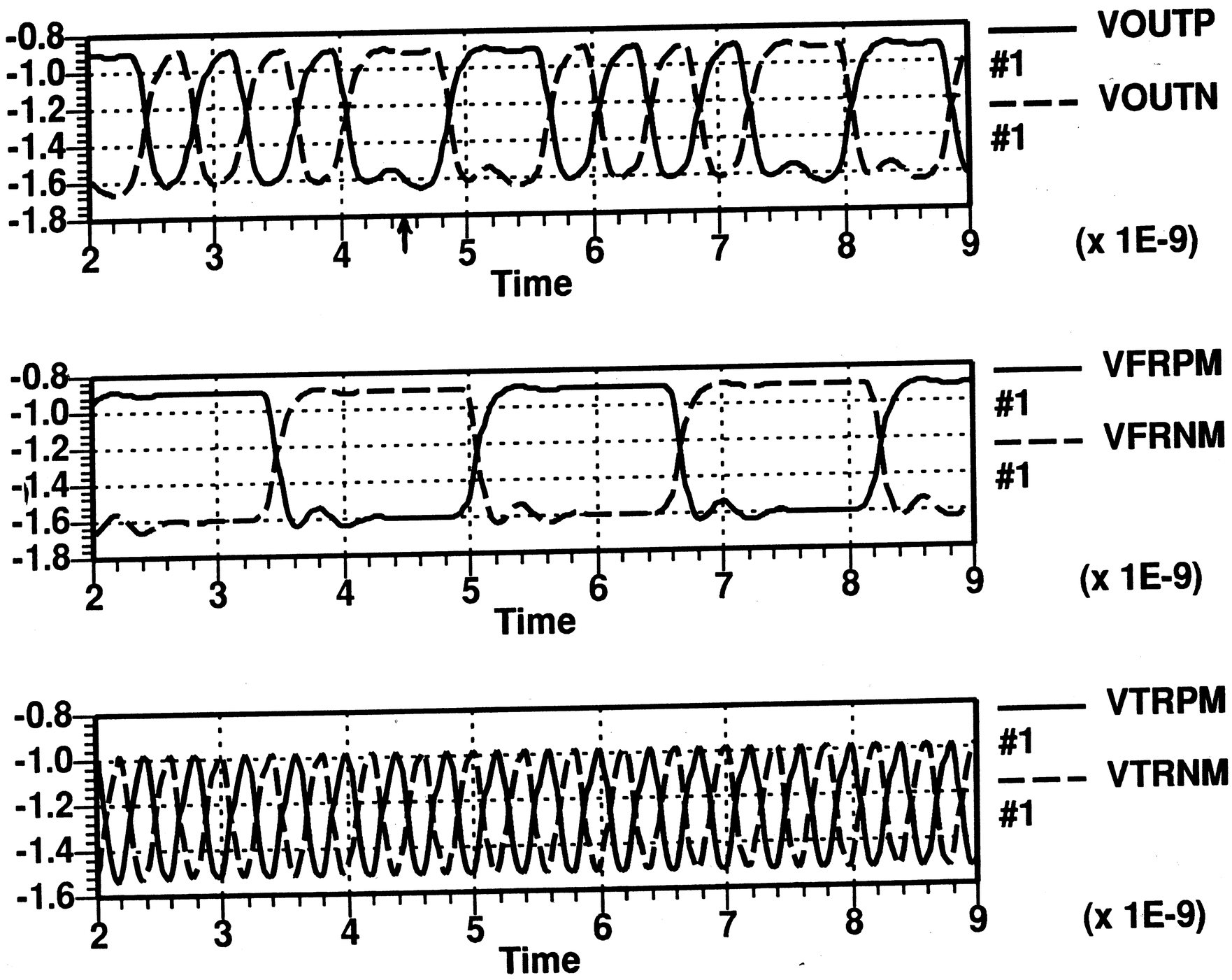


Figure 39. 2.5GBit/s typical output waveforms for Multiplexer

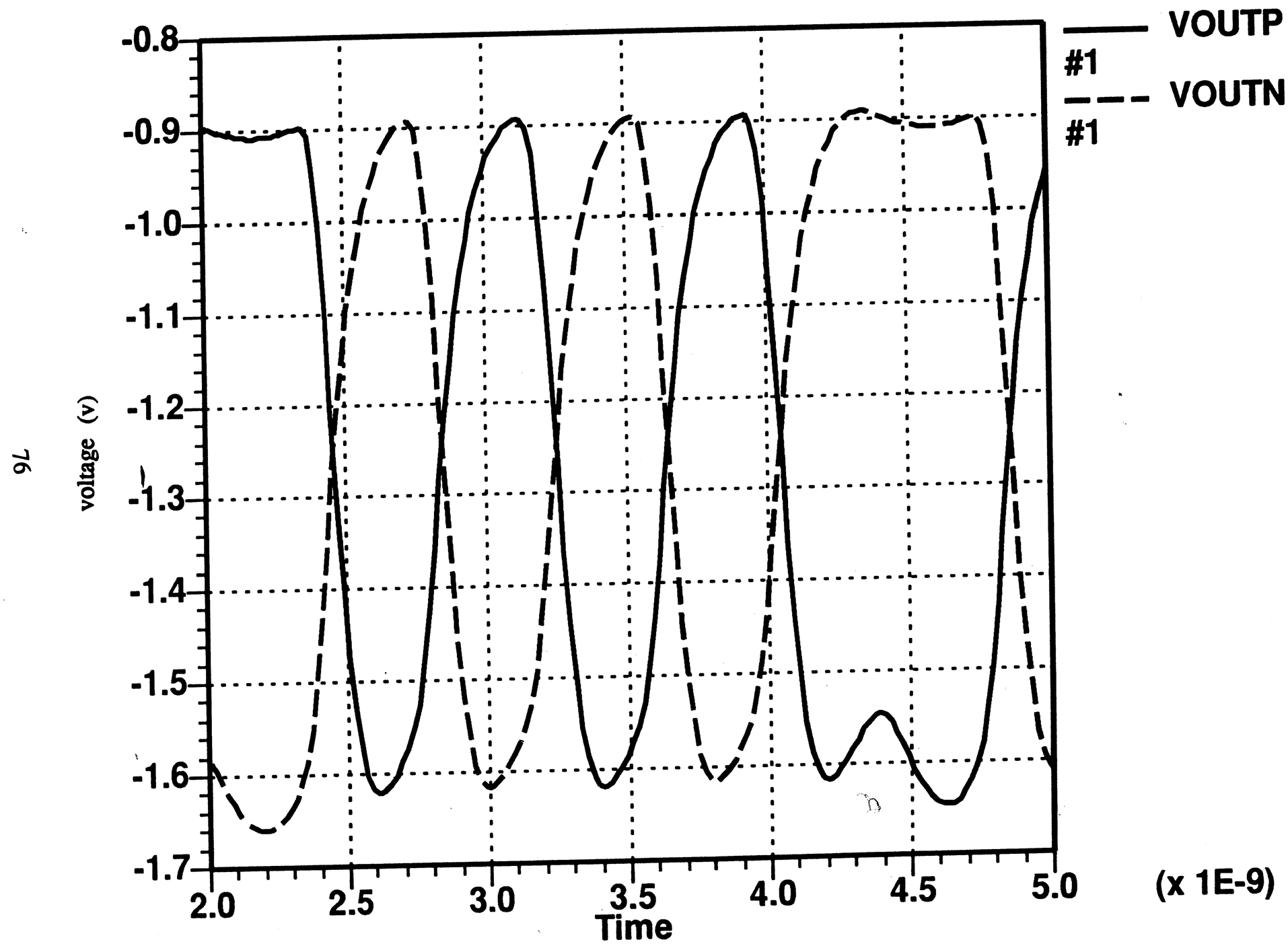


Figure 40. Exploded view of 2.5GBit/s typical Multiplexer output data

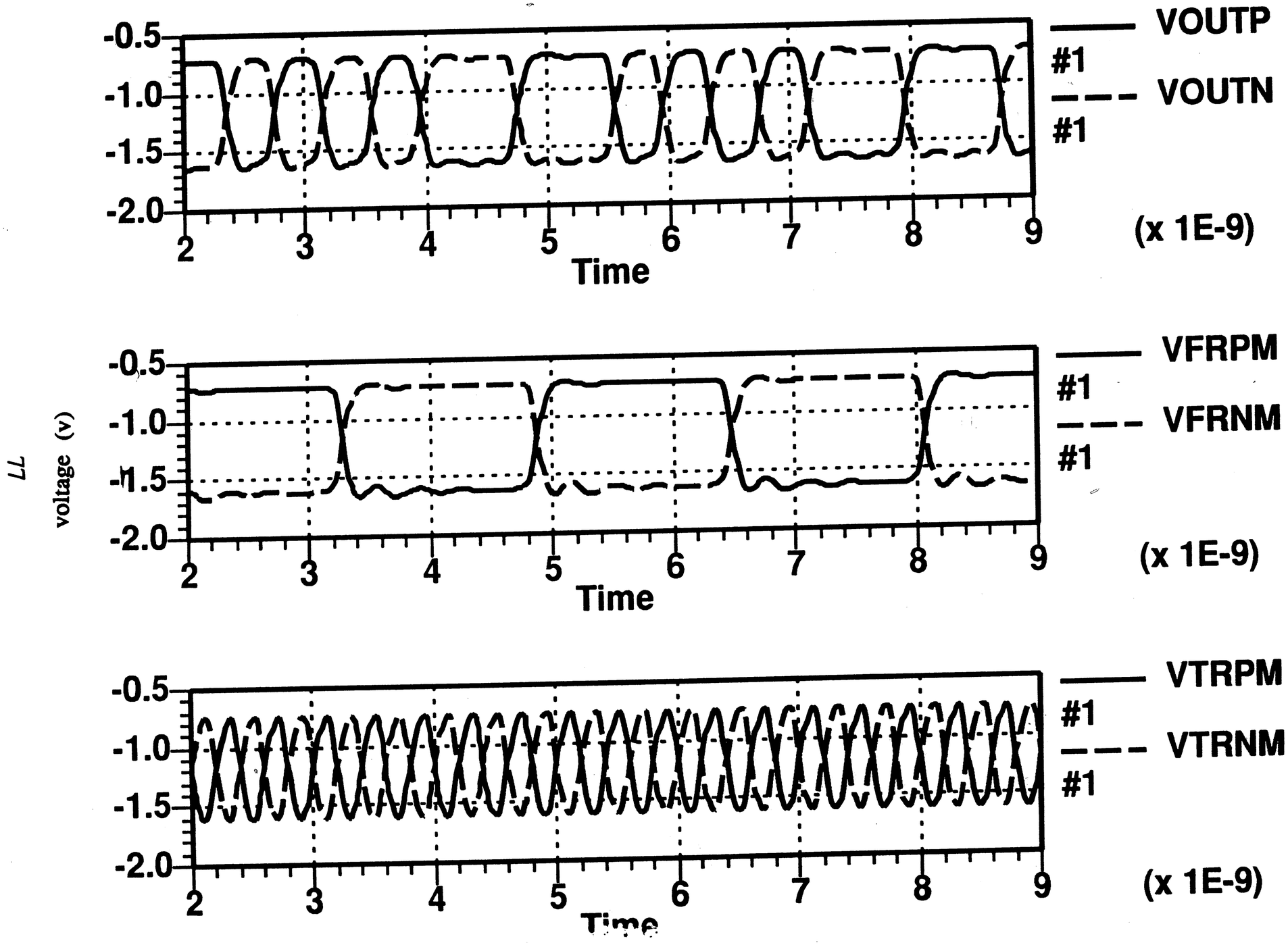


Figure 41. 2.5GBit/s worst case fast output waveforms for Multiplexer

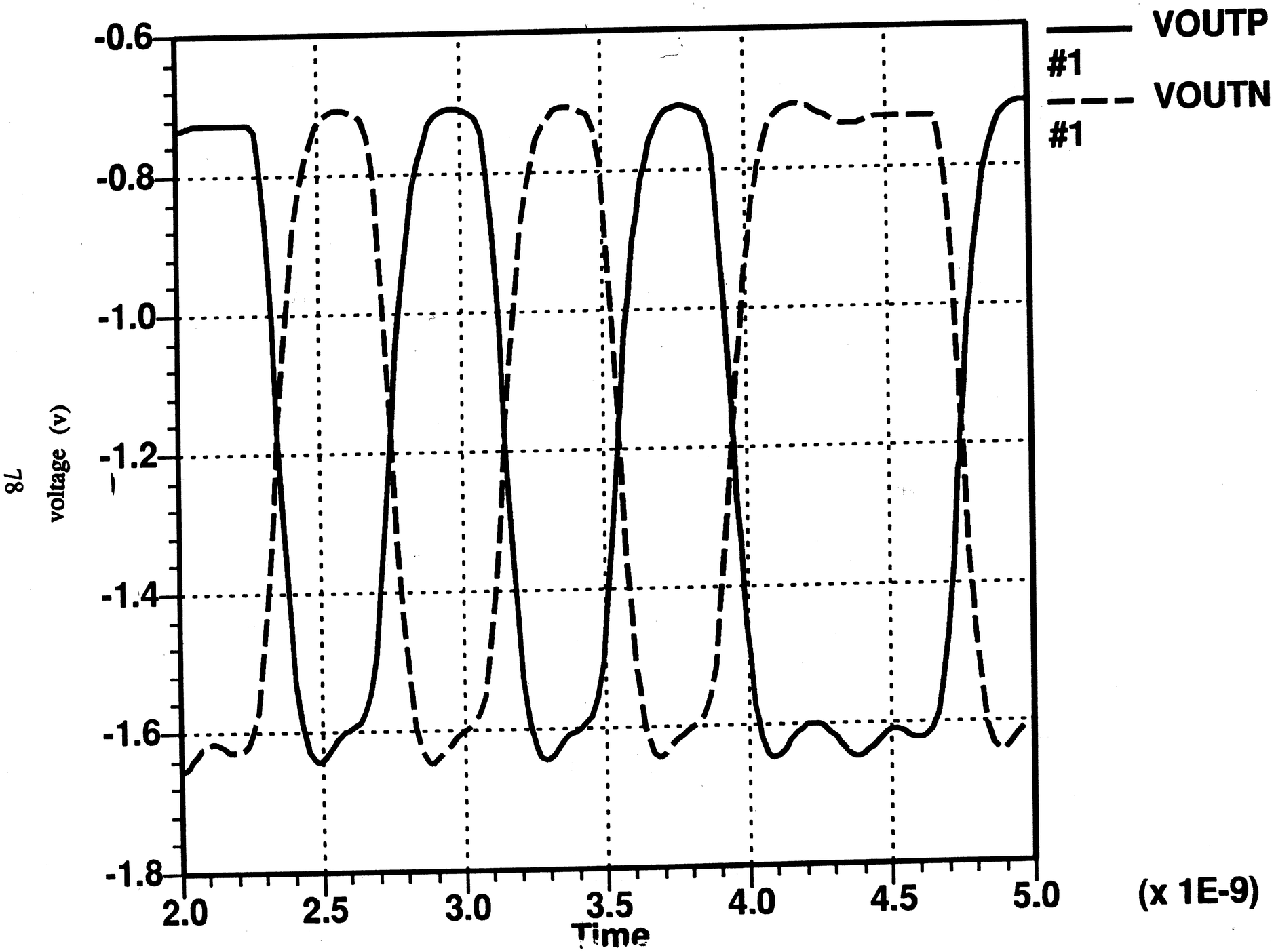


Figure 42. Exploded view of 2.5GBit/s worst case fast Multiplexer output data

ADVICE 1P AS OF 091489 RUN ON 04/15/90 AT 00:00:13 S# 26439
(0. DEG C) mux worse case slow 2.5GBit/s output

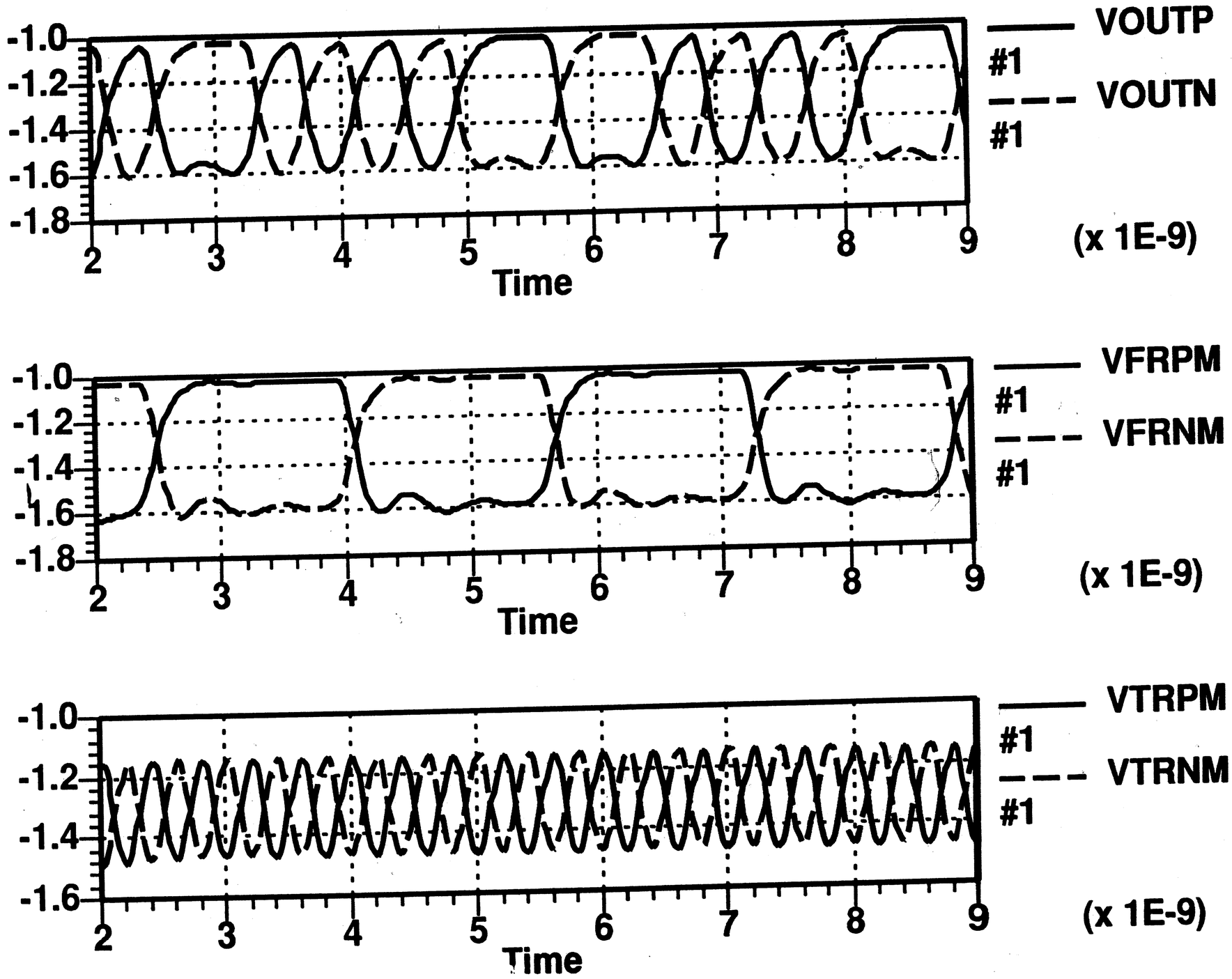


Figure 43. 2.5GBit/s worst case slow output waveforms for Multiplexer

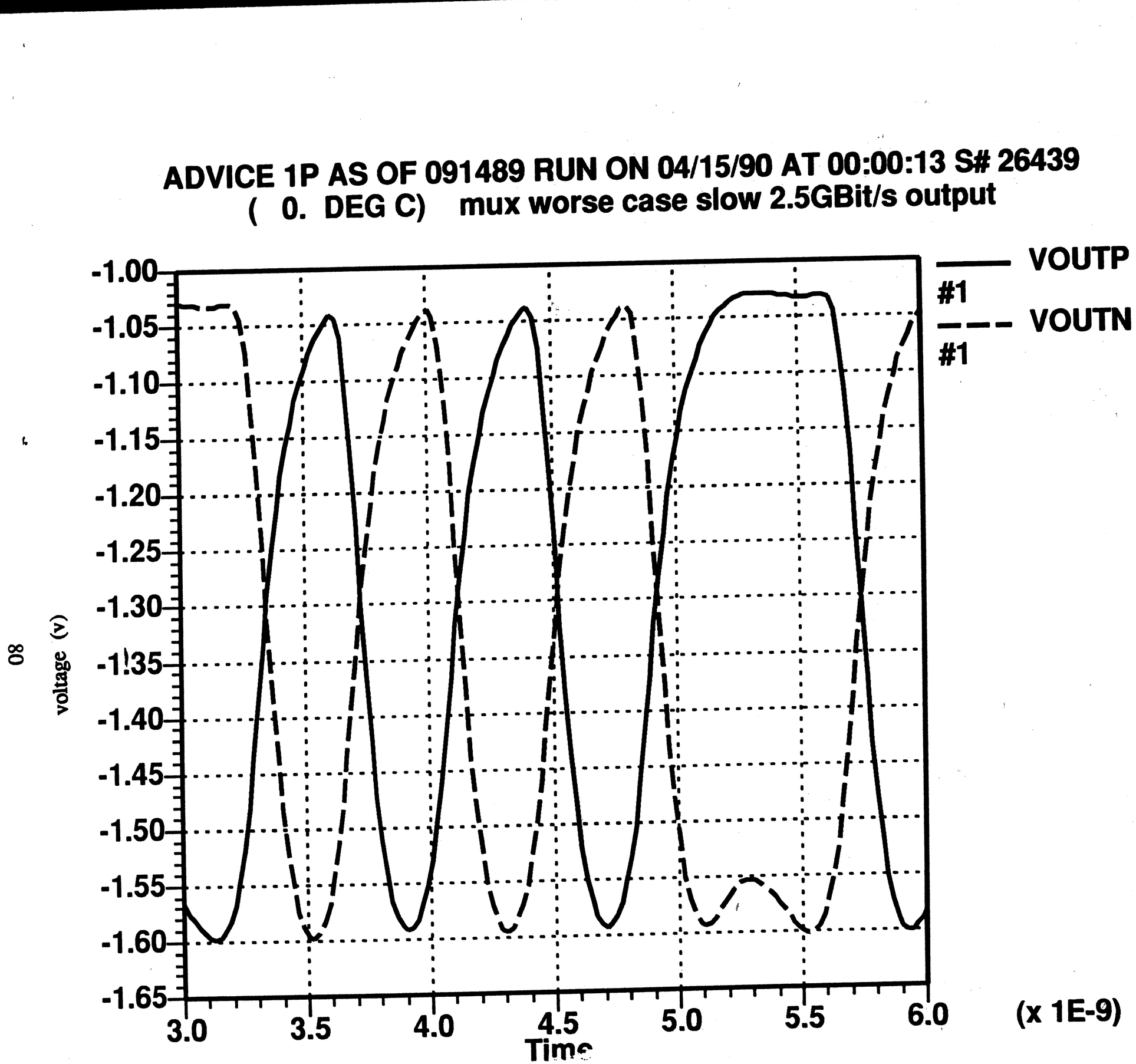


Figure 44. Exploded view of 2.5GBit/s worst case slow Multiplexer output data

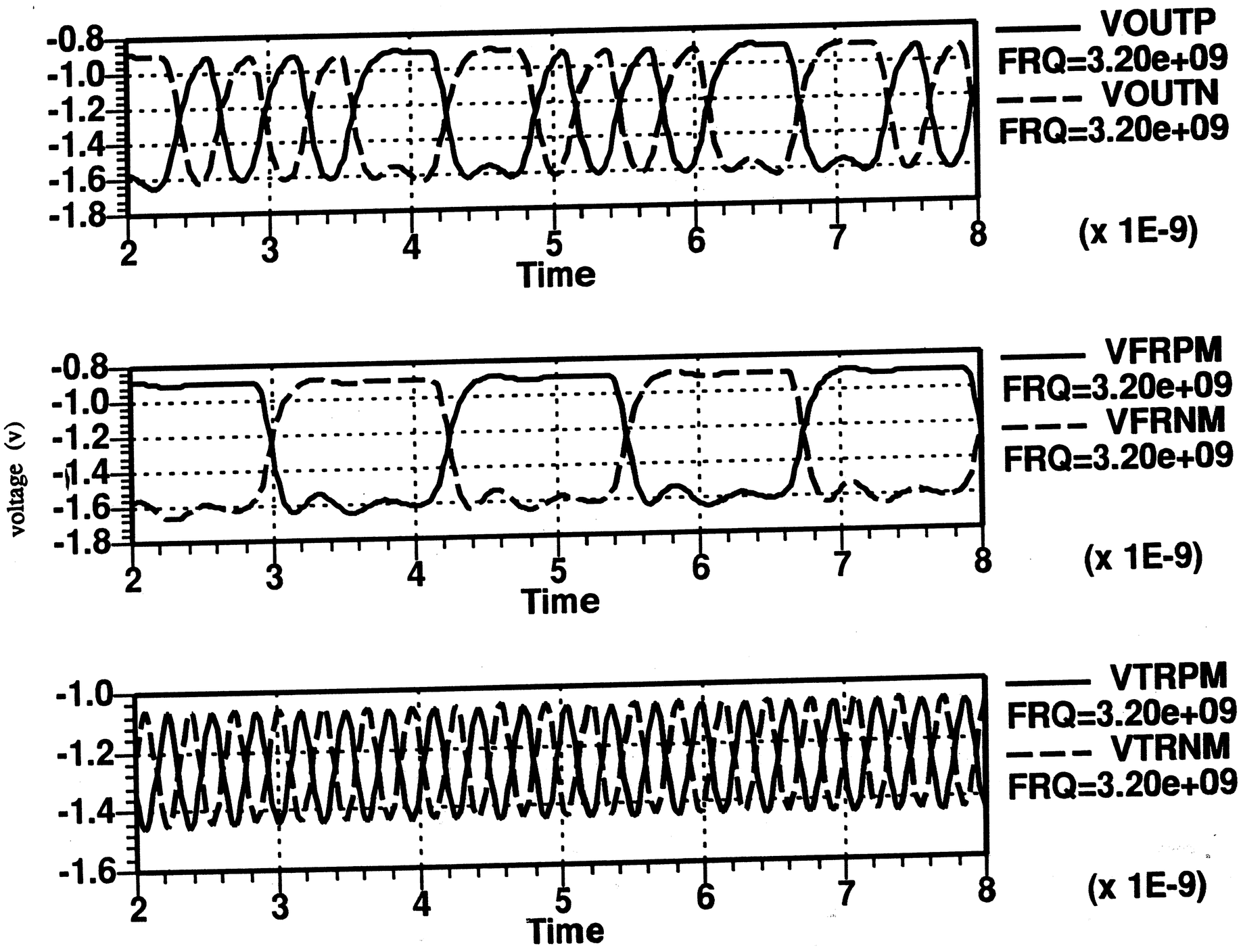


Figure 45. 3.2GBit/s typical output waveforms for Multiplexer

the 3.2Gbit/s peak typical frequency that the circuit is expected to operate. Above this frequency, the first stage of the ripple counter fails to toggle correctly and shuts down the operation of the entire circuit.

The simulation results for the Time Division Demultiplexer are shown in figures #46 - #51. First, figure #46 shows four bytes of information output from data0-data7. These bytes are a demultiplexed version of the 2.5Gbit/s pseudo random waveform 01011001101001001111l. The first logic low bit is output from data6 during byte "A" at 5ns, followed by the second logic high bit on output data7. The next 8 bits of information 01100110 are output during byte "B" at 8ns on data channels 0 through 7. Likewise, this byte of information is followed by byte "C" at 11.2ns consisting of the next 8 data bits 10010011. Finally, the last of the information arrives in byte "D" at 14.4ns. If the system designer had defined the first logic low bit as bit 0 of byte "A", then a simple shift of two could be done by taking input SHIFT2 high during an input active clock cycle, and the data would be shifted in time and space and realigned with its proper output channel. Figure #47 demonstrates the skew results between all of the output channels and the frame clock, by overlaying all of the signals with the frame clock output FRPDM. Note, that this figure is an exploded view of the previous figure #46 between 11 and 11.8ns. This point was chosen because it possessed the highest number of simultaneous output transitions. In total, 6 of the 8 outputs changed state during this time period. From the figure, the amount of skew between all of the data channels is undetectable and there is less than 200psec skew between the frame clock and the data channels. This is considerably less than the system requirement of 2 nanoseconds. Figure #48 demonstrates the Demultiplexer's performance under worst case fast conditions. The circuit functions exactly as it had under typical conditions by demultiplexing the input 2.5Gbit/s pseudo random waveform to the 8 output data channels. Again, a simple shift of 2 can be performed to realign the data if necessary. Measurement of the output data channel and frame clock still shows undetectable skew which is still undetectable between the data channels. Furthermore, the skew between the frame clock and the output data channels is reduced to almost 100psec which is still considerably less than the system requirement. Figure #49 demonstrates the circuits performance under worst case slow conditions. The results are identical to those under worst case fast conditions with still undetectable data output skews and 200psec frame clock to data output skew. Finally, figures #50 and #51 demonstrate 3.1Gbit/s peak typical circuit

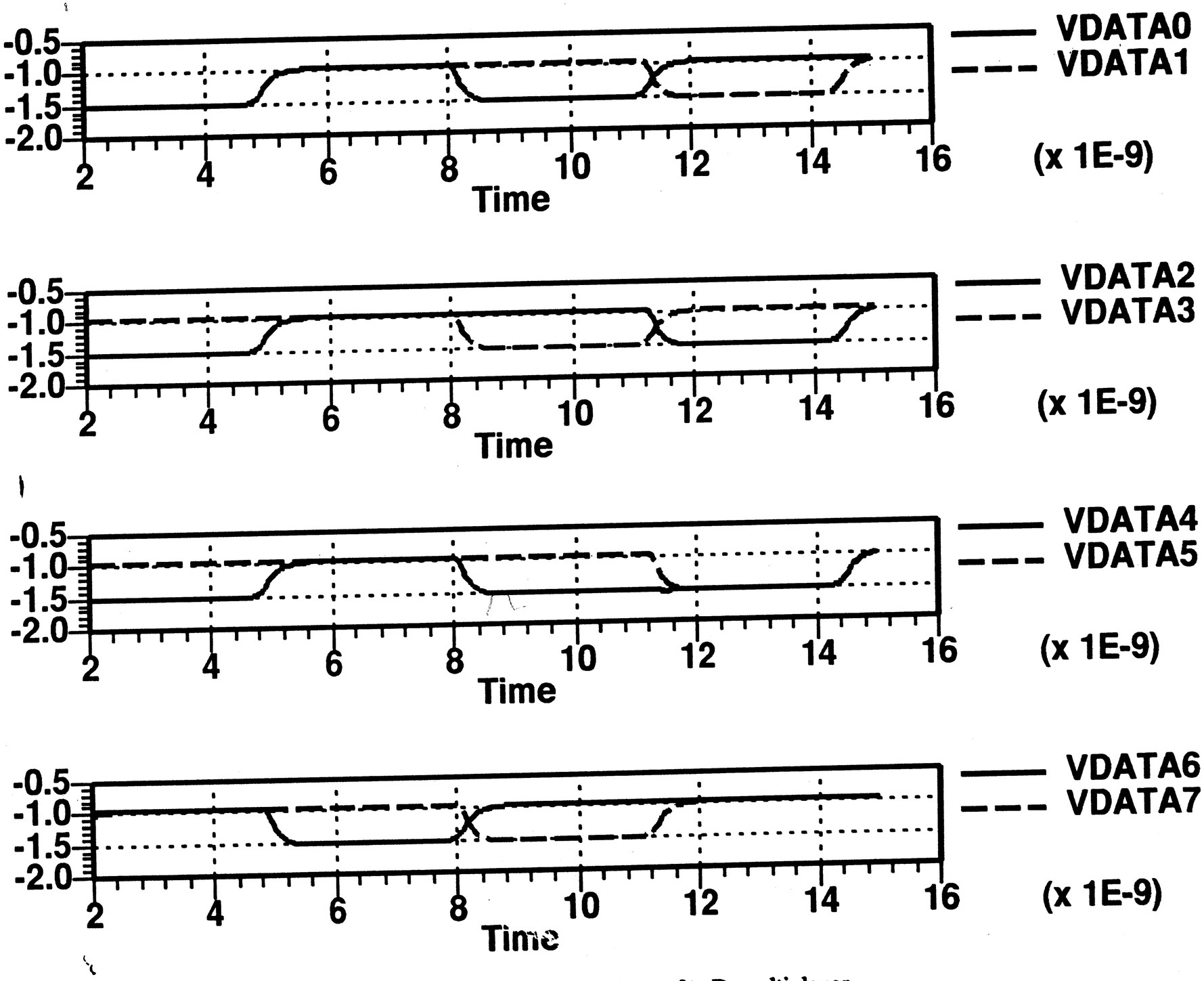


Figure 46. 2.5GBit/s typical output waveforms for Demultiplexer

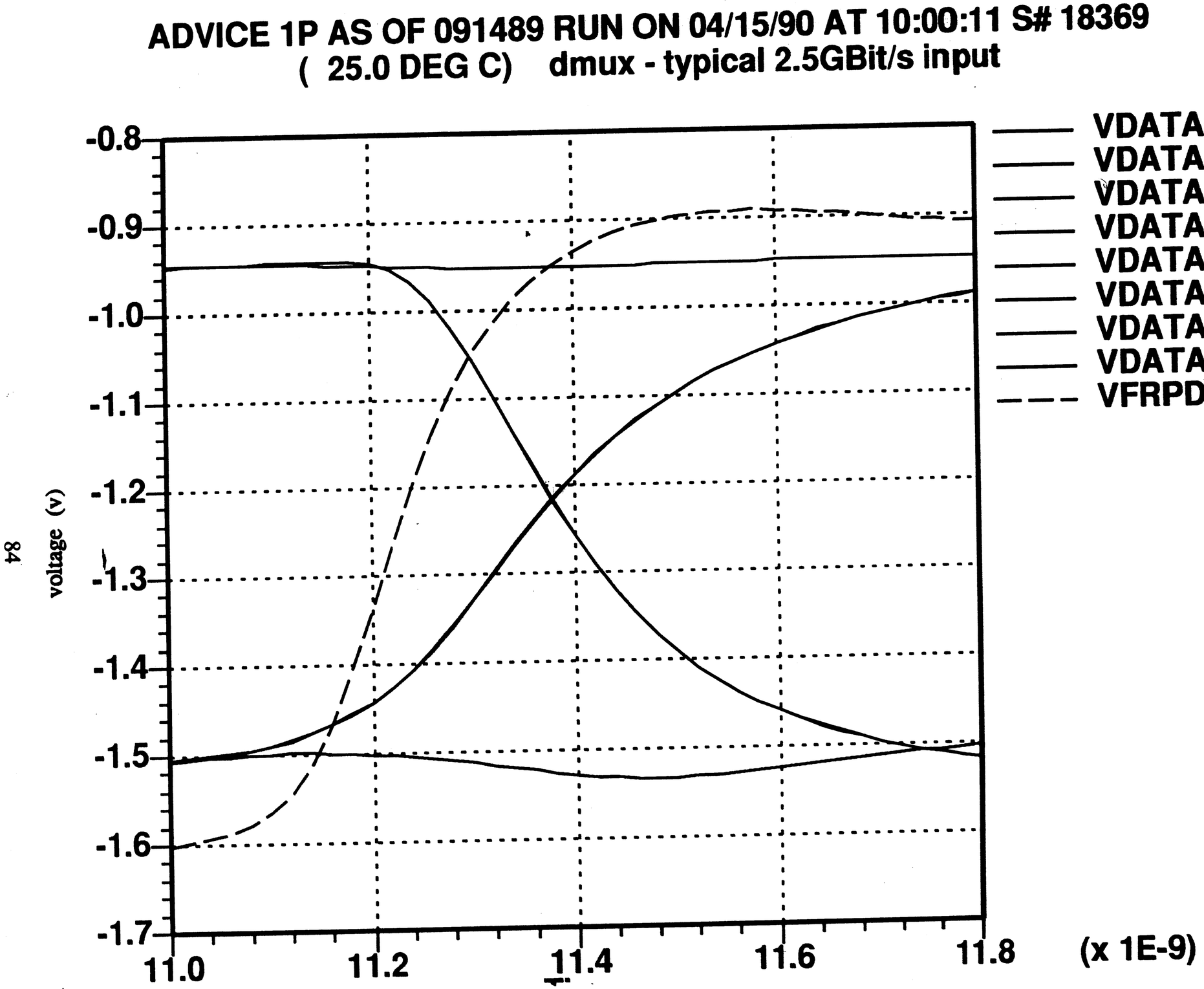


Figure 47. 2.5GBit/s typical skew results for Demultiplexer outputs

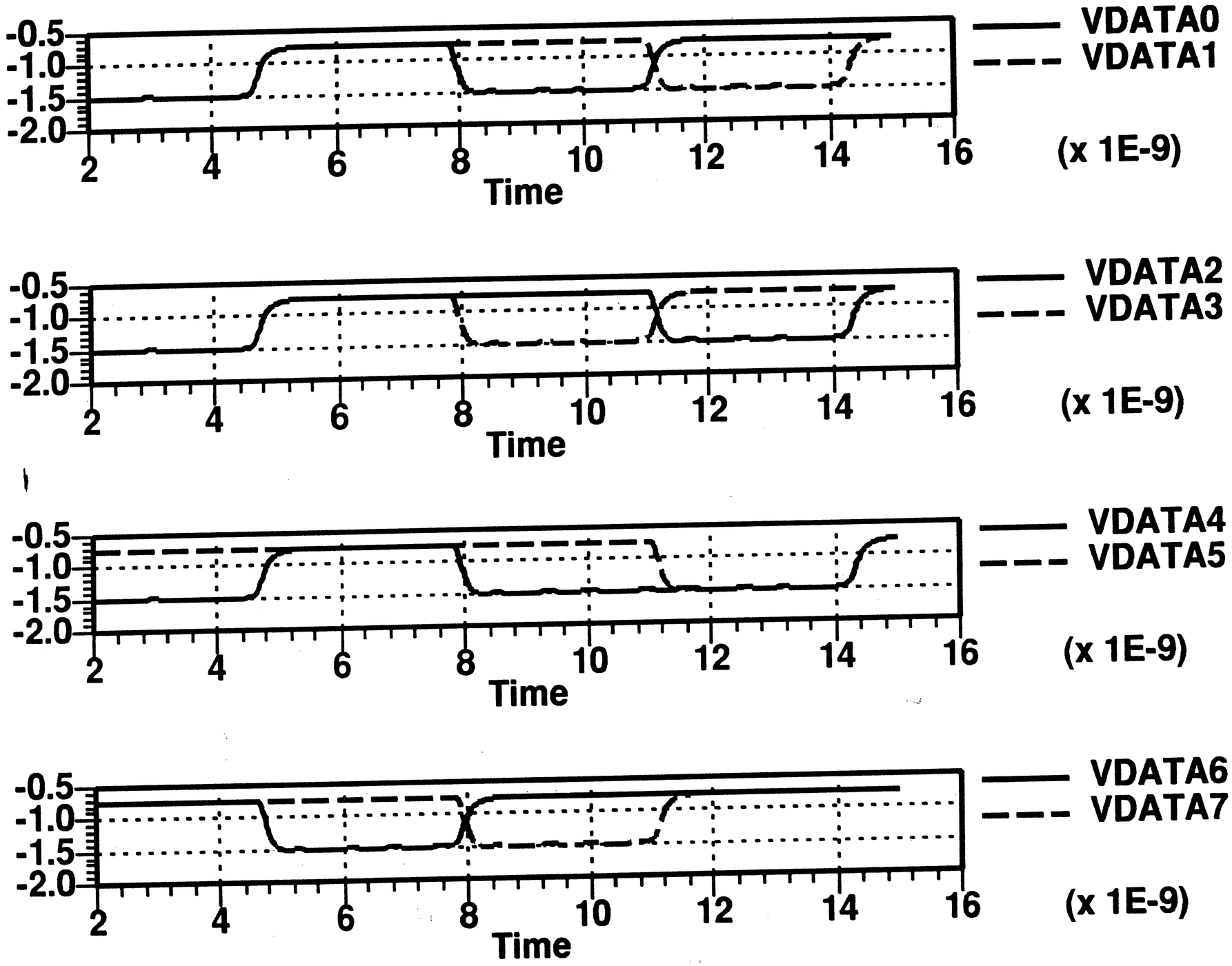


Figure 48. 2.5GBit/s worse case fast output waveforms for Demultiplexer

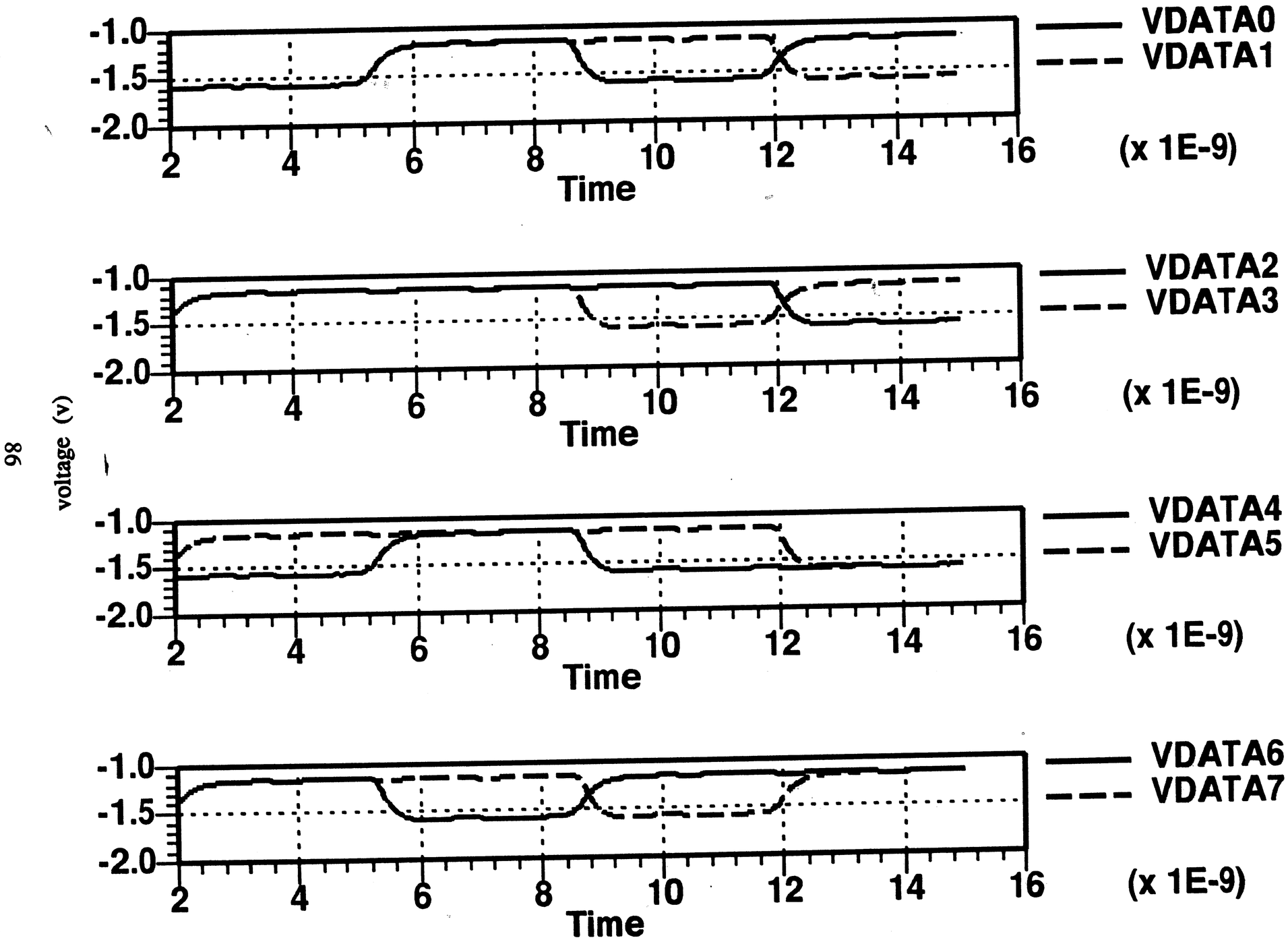
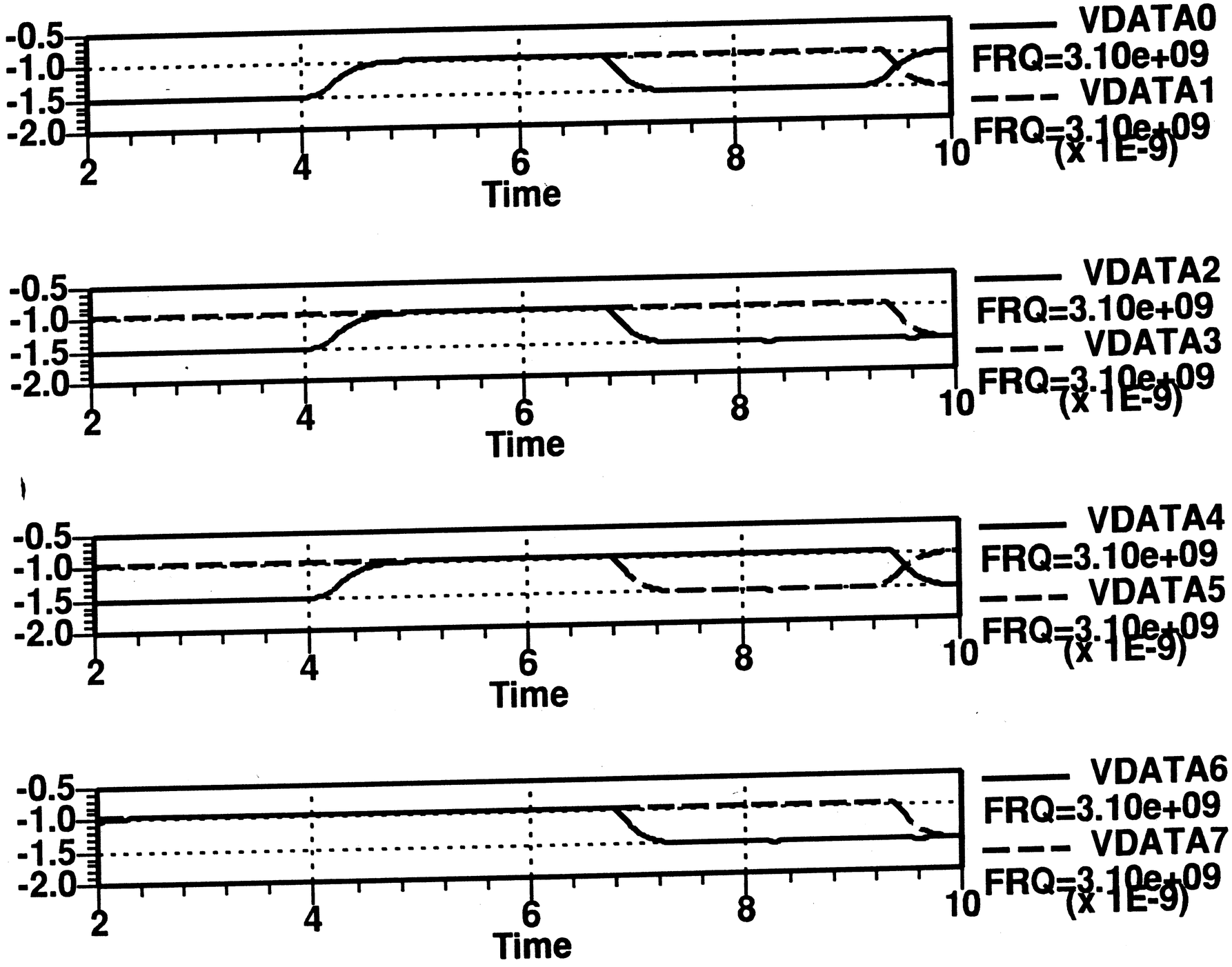


Figure 49. 2.5GBit/s worse case slow output waveforms for Demultiplexer



voltage (v)

87

Figure 50. 3.1GBit/s typical output waveforms for Demultiplexer

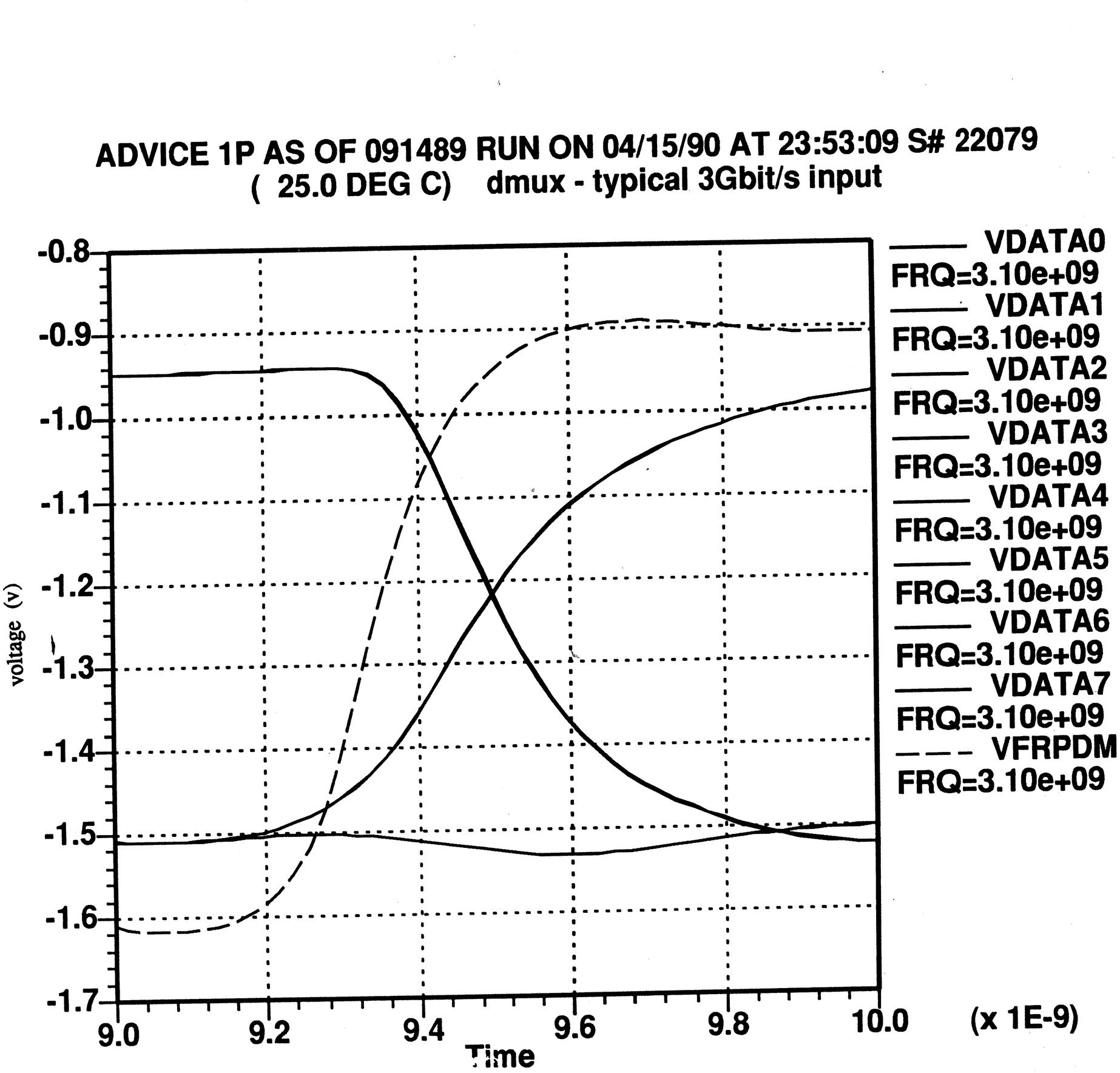


Figure 51. 3.1GBit/s typical skew results for Demultiplexer outputs

performance. Above this frequency, the first stage of the ripple counter stops functioning.

10. Conclusion

This paper has described the design and layout of a Silicon Integrated Circuit consisting of a Bit Interleaving Time Division Multiplexer and Time Division Demultiplexer fabricated in AT&T's BEST-1 (Bipolar Enhanced Self-aligned Technology) process. The circuits follow a ripple counter controlled architecture first presented by M. Ida, et al. in August of 1989 for Nippon Telephone and Telegraph (NTT) in a 16 to 1 channel and 1 to 16 channel Time Division Multiplexer and Demultiplexer fabricated in the Gallium Arsenide Technology. This work has managed to improve on the NTT design by adding a provision to the Demultiplexer for the shifting of the output data in time and space such that input bits of information can be realigned to their appropriate output data channel amidst the proper byte of information. Furthermore, in comparison to an equivalent 8 to 1 channel and 1 to 8 channel NTT design, this work consumes 13% less power by using 18 fewer data latches. This architecture was selected after the completion of a literature review which showed only one other common Time Division Demultiplexer Architecture and two other Multiplexer architectures. The Time Division Demultiplexer architecture followed a shift register approach, where the input data bit stream is serially shifted into an "N" bit shift register which in turn controlled the parallel output of data onto the "N" output data channels once the shift register was filled. After dumping, the cycle repeats until the register is again filled. One of the Multiplexer architectures investigated also followed this same approach, by parallel loading "N" input data channels into a "N" bit shift register, and then serially shifting the data bits off chip one bit at a time. Both of these are viable approaches and will meet all of the system requirements and specifications discussed in chapter 2, but the frequency at which they will operate is limited by the high fanout loading of the input high speed clock which must drive each of the flip-flops of the "N" bit shift register. The second Multiplexer architecture improves upon this frequency problem by using a retiming approach of the high speed clock to remove the fanout problem. Thus, the high speed clock merely drives a 2 bit Johnson counter which when decoded and fed into a shift register, controls the select leads to an asynchronous multiplexer. Meanwhile, a version of the clock edge which propagates through the counter and the shift register is equally delayed through a series of gates in order that the data which is selected can be reclocked by a final flip-flop before being transmitted. Although this architecture improves upon the clock fanout issue, it creates a new issue regarding the accuracy that the delay line can be controlled

through manufacturing.

The ripple counter architecture selected for this work does not have the fanout issue of the high speed clock, nor does it have any critical timing issues that will vary from die lot to die lot. This architecture is controlled by a simple 3 bit ripple counter, the first stage of which is clocked by the high speed clock. In the Time Division Multiplexer, the outputs from this ripple counter control the select and enable inputs to a cascade of 2 to 1 channel data latches. The select and enable scheme is set up such that the enabling of the data to the latch is done at twice the frequency that the data is selected. This way, each Mux data latch manages to combine two data channels into one. Therefore, the first stage of the multiplexer which consists of four mux data latches, combines the 8 input data channels into 4. Likewise, the second multiplexing stage which consists of two mux latches combines the four data channels to two. Finally, a mux flip-flop is used in the last stage to multiplex the two channels to one before the data is serially transmitted from the circuit. The Time Division Demultiplexer operates in reverse of the Multiplexer by cascading demultiplexing stages which consist of a D-type flip-flop and data latch. The inputs of these two gates are driven in parallel by the input serial data stream, and they are clocked by a 1.25GHz clock created by the first stage of the ripple counter. On positive edges of this clock, even bits of data and taken from the data stream by the D-type flip-flop and are channeled to a second demultiplexing stage, and during logic low conditions of this clock, the odd bits of data are directed through the data latch to a separate demultiplexing stage. Both of these second demultiplexing stages are then controlled by the same 600MHz clock output from the ripple counter. These stages repeat the demultiplexing function performed by the first stage and create four data channels out of the two that were input to them. These four data channels are then directed into the next demultiplexing stage which consists of 4 parallel stages equivalent to the first stage, except the data latch is replaced by a three stage flip-flop consisting of a data latch in series with of a D-type flip-flop. This is done in order that the final output of data can occur in a parallel fashion with minimal skew between data channels. In effect, the three stage data latch delays the output of its data bit until the D-type flip-flop is ready to output its bit which occurs on the positive clock edges of the 312MHz clock. Furthermore, with the addition of the SHIFT1-SHIFT4 inputs, which either enable or disable the toggling of each bit of the ripple counter, the output data can be redirected to different output data channels in order that the proper bit of

data arrives at the correct data channel output during the correct byte of information.

In order to achieve the high speed data bit rates necessary for the operation of this circuit, these designs were implemented in a full differential ECL approach with a 400mV internal logic signal swing. This paper discussed the migration from the single-sided amplifier to the differential ECL inverter, and how a stacked logic approach was used to build the eight internal and five external input and output buffer building blocks necessary to perform the required logic. The layout of these blocks followed a standard cell internal and external template approach, which were then incorporated into an overall standard cell/gate array die design.

Finally, this paper concluded with the presentation of the simulation results showing circuit operation which met all of the system requirements and specifications listed in chapter 2. The Time Division Multiplexer easily achieved the 2.5Gbit/s output data bit rate over worst case fast, slow and typical processing. Furthermore, it produced pulses which had typically less that 7% edge jitter. This virtually jitter-free output data-bit-stream also was shown to exist at 3.2Gbit/s operation, which is the highest frequency at which the Multiplexer will operate before the first stage of the ripple counter stops functioning. The Time Division Demultiplexer showed equally impressive results with undetectable skew between output data channels at both 2.5Gbit/s and 3.1Gbit/s input data rates. Furthermore, the skew between the data channels and the rising edge of the frame clock is three orders of magnitude less than the 2 nanoseconds allotted in the system specifications.

This Integrated Circuit comes at a time when many system designers are questioning the short term future of Gallium Arsenide circuits. Many Semiconductor houses are struggling with the high development expenditures for this technology which is riddle with many processing difficulties. This Time Division Multiplexer and Demultiplexer are designed in a viable silicon based process. Furthermore, upon analyzing the speed power graph in figure #52, originally presented by M. Ida, et al. in the IEEE Journal of solid state circuits in August 1989, these two circuits presented in this paper clearly top the charts of known Time Division Multiplexers and Demultiplexers second only to the 16-to-1 and 1-to-16 channel circuits by NTT.

Continuing research and design with the Bit Interleaving Multiplexer design described in this paper could lead to a circuit which could operate at over 6Gbit/s by replacing the flip-flop in the final stage of the Multiplexer with a mux data latch. As stated in chapter 6, this would

SPEED POWER PERFORMANCE OF KNOWN GALLIUM

ARSENIDE AND SILICON MUX'S and DMUX'S

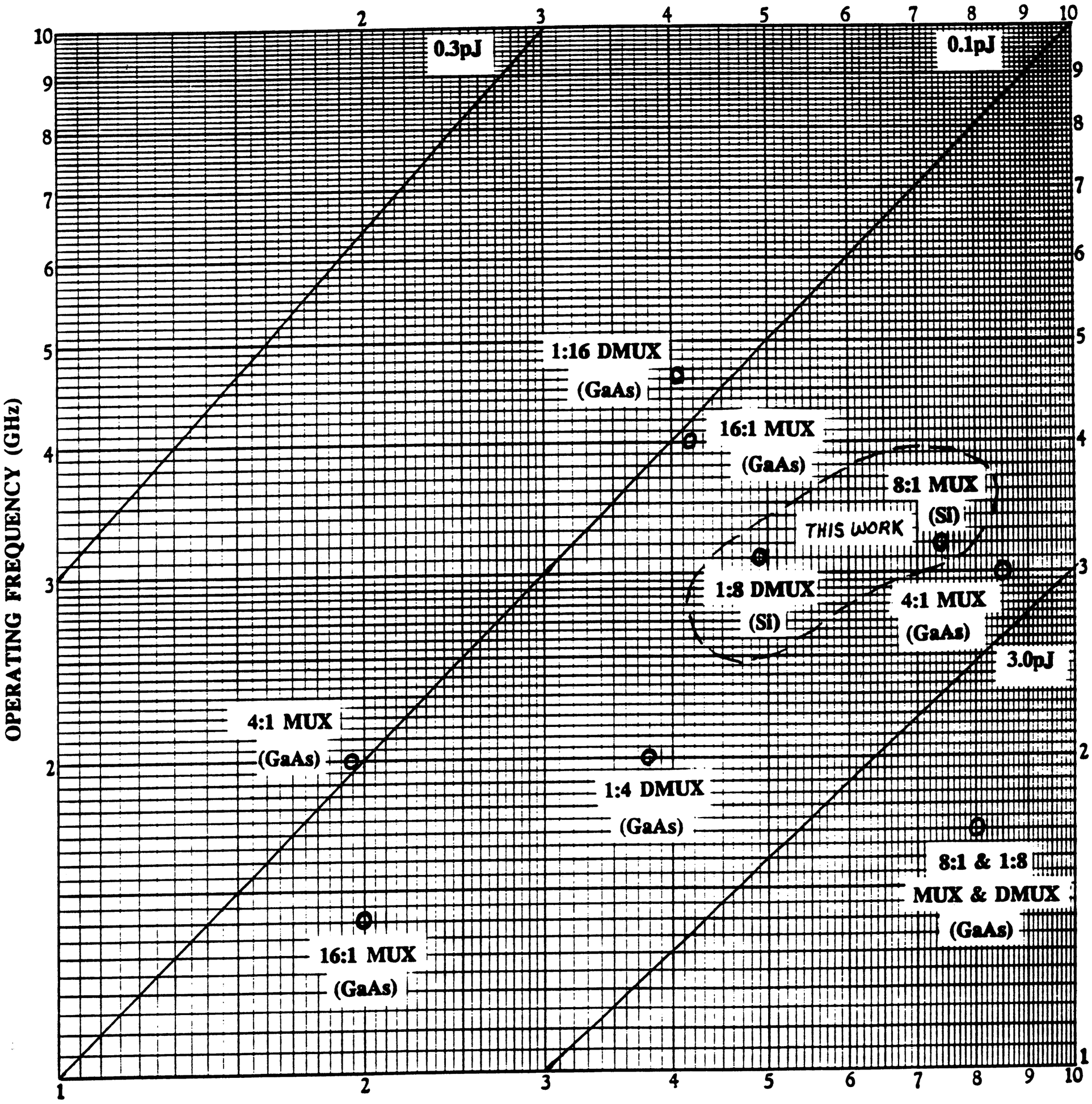


Figure #52 AVERAGE DISSIPATED POWER PER GATE (milliwatts/gate)

require a very accurate internal high speed clock with controlled 50% duty cycles in order to manage edge jitter of the output data bit stream. The author, at this time, believes that this circuit performance and control can be achieved, and with this advancement along with the further processing enhancements projected in BEST-2, silicon circuit performance close to 10Gbit/s operation may be realizable.

1. M. Ida, N. Kato, and T. Takada, "A 4-GBits/s GaAs 16:1 Multiplexer/1:16 Demultiplexer LSI Chip,"*IEEE Journal of Solid-State Circuits*, Vol. 24, No. 4, August 1989, pp. pp. 928-932.

2. S. K. Ghandhi, *VLSI Fabrication Principles, Silicon and Gallium Arsenide*, John Wiley & Sons, New York, New York, 1983. pp. 5, 12-14.

3. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, Vol. 2, John Wiley & Sons, New York, New York, 1984. pp. 12-13, 32-34, 59-72, 196.

4. D. K. Ferry, L. A. Akers, and E. W. Greeneich, *Ultra Large Scale Integrated Microelectronics*, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1988. pp. 146, 170-172.

5. T. Chiu, G. M. Chin, M. Y. Lau, R. C. Hanson, M. D. Morris, K. F. Lee, A. M. Voshchenkov, R. G. Swartz, V. D. Archer, and M. T. Y. Liu&', "Nonoverlapping Super Self-Aligned Device Structure for High-Performance VLSI,"*IEEE Electron Device Letts*, Vol. 11, No. 2, February 1990, pp. 85-87.

6. R. G. Swartz, "Ultra-High Speed Bi-Phase Multiplexer/ Demultiplexer Architectures for Optical Communications,"*11365-860923-21TM*, September 23, 1986.

7. G. Flower, "A 3 Gigaherz 4:1 Time Division Multiplexer with Output Retiming,"*IEEE 1988 Bipolar Circuits & Technology Meeting*, paper 7.3, 1988, pp. pp. 146-149.

8. A. Microelectronics, *BEST-1 Series High-Performance ECL Gate Arrays*, Data sheet, AT&T Microelectronics, Allentown, Pa 18103, 1990. DS89-143DBIP.

9. W. R. Griesbach, *A High Speed, Low Power Bipolar Static Random Access Memory Featuring ECL I/O and a Collector-Coupled Memory Cell*, Master's Thesis, Lehigh University, 1988. pp. 31.

10. F. J. Widlar, "New Developments in IC Voltage Regulators,"*ISSCC Digital Technical Papers*, February 1970, pp. 157-159.

11. Motorola, *MCA3 Series Third Generation ECL Macrocell Arrays*, Data Sheet,

Motorola, 1988. pp. 7-8.

12. TriQuint, *68 Lead High Speed Ceramic Package Data Sheet*, Product Advisory, TriQuint Semiconductor, May 23, 1989.

11. Appendices

Advice Run Files[1]

```
**********************************************************************
*this file represents the electrical connectivity for the
*Bit interleaving Time Division Multiplexer
**********************************************************************
***************    Audit Summary for   MUXCTR    ****************
* Subcircuit mode audits were NOT performed on this single page.
* **************    SCHEMA     Version:2.3.0    E: 2.0.0    **************
*
*PAGE: MUXCTR.1    DATE: 89 Nov 16 10:16:36
*
   xibg1  (vbbx1,vbby1,vcs1)  ibg ;*handles buffers
   xibg2  (vbbx2,vbby2,vcs2)  ibg ;*handles top row
   xibg3  (vbbx3,vbby3,vcs3)  ibg ;*handles bottom row
*top row of logic
   XMFF1 (vbbx2,vbby2,vcs2,q1xnm,q1xpm,clrxpm,clrxnm,clrxpm,clrxnm,ckpm,cknm
+          :q1xpm,mcky2p,q1xnm,mcky2n)  f1dchd2
   XMFF2 (vbbx2,vbby2,vcs2,q2xnm,q2xpm,clrxpm,clrxnm,mcky2p,mcky2n
+          :q2xpm,mcky4p,q2xnm,mcky4n)  F1DCHD
   XMFF3 (vbbx2,vbby2,vcs2,q3xnm,q3xpm,clrxpm,clrxnm,mcky4p,mcky4n
+          :q3xpm,mcky8p,q3xnm,mcky8n) F1DCHD
   xmff4 (vbbx2,vbby2,vcs2,mevenp,mevenn,moddp,moddn,mcky2p,mcky2n,ckpm,cknm
   +      :mux0t7p,mux0t7n) f2dahd
   XBUF2M (vbbx2,vbby2,vcs2,q3xpm,q3xnm:framep,framen) RCR1HD
*bottow row of logic
   xml1 (vbbx3,vbby3,vcs3,id0xp,id0xn,id4xp,id4xn,mcky8p,mcky8n,mcky4n,mcky4p
   +      :mux04p,mux04n) f4dahd
   xml2 (vbbx3,vbby3,vcs3,id2xp,id2xn,id6xp,id6xn,mcky8p,mcky8n,mcky4n,mcky4p
   +      :mux26p,mux26n) f4dahd
   xml3 (vbbx3,vbby3,vcs3,id1xp,id1xn,id5xp,id5xn,mcky8p,mcky8n,mcky4n,mcky4p
   +      :mux15p,mux15n) f4dahd
   xml4 (vbbx3,vbby3,vcs3,id3xp,id3xn,id7xp,id7xn,mcky8p,mcky8n,mcky4n,mcky4p
   +      :mux37p,mux37n) f4dahd
   xml5 (vbbx3,vbby3,vcs3,mux04p,mux04n,mux26p,mux26n,mcky4p,mcky4n,mcky2n
   +      ,mcky2p:mevenp,mevenn) f4dahd
   xml6 (vbbx3,vbby3,vcs3,mux15p,mux15n,mux37p,mux37n,mcky4p,mcky4n,mcky2n
   +      ,mcky2p:moddp,moddn) f4dahd
*buffers
   XBCLKM (vbbx1,vbby1,vcs1,clkpm,clknm:ckpm,cknm) EBCB
   XBclrm (vbbx1,vbby1,vcs1,clrm:clrxpm,clrpy,clrxnm,clrny) EI1N1
   xbdata0 (vbbx1,vbby1,vcs1,mdata0:id0xp,d0y,id0xn,d0yn) ei1n1
   xbdata4 (vbbx1,vbby1,vcs1,mdata4:id4xp,d4y,id4xn,d4yn) ei1n1
   xbdata2 (vbbx1,vbby1,vcs1,mdata2:id2xp,d2y,id2xn,d2yn) ei1n1
   xbdata6 (vbbx1,vbby1,vcs1,mdata6:id6xp,d6y,id6xn,d6yn) ei1n1
   xbdata1 (vbbx1,vbby1,vcs1,mdata1:id1xp,d1y,id1xn,d1yn) ei1n1
   xbdata3 (vbbx1,vbby1,vcs1,mdata3:id3xp,d3y,id3xn,d3yn) ei1n1
   xbdata5 (vbbx1,vbby1,vcs1,mdata5:id5xp,d5y,id5xn,d5yn) ei1n1
   xbdata7 (vbbx1,vbby1,vcs1,mdata7:id7xp,d7y,id7xn,d7yn) ei1n1
   xbout (vbbx1,vbby1,vcs1,mux0t7p,mux0t7n:outp,outn) xbmb
   XBTRM (vbbx1,vbby1,vcs1,ckpm,cknm:trpm,trnm)    xbmb
   xbfrm (vbbx1,vbby1,vcs1,framep,framen:frpm,frnm)      xbmb
*fanout loading
   XLOAD1a (trpm) ECL50
   XLOAD1b (trnm) ECL50
   xload2a (outp) ECL50
   xload2b (outn) ECL50
   xlo3a   (frpm) ECL50
   xlo3b   (frnm) ECL50
*load current sources
   XLOAD3 (mcky8n,vcs2) I1MA
   XLOAD3a (mcky8n,vcs2) I550UA
   XLOAD4 (mcky8p,vcs2) I1MA
   XLOAD4a (mcky8p,vcs2) I550UA
```

```
XLOAD5 (mcky4n,vcs2) I1MA
XLOAD5a (mcky4n,vcs2) I550UA
XLOAD6 (mcky4p,vcs2) I1MA
XLOAD6a (mcky4p,vcs2) I550UA
XLOAD7 (mcky2n,vcs2) I1MA
XLOAD7a (mcky2n,vcs2) I550UA
XLOAD8 (mcky2p,vcs2) I1MA
XLOAD8a (mcky2p,vcs2) I550UA
xloa0 (framep,vcs2) I550UA
xloa1 (framen,vcs2) I550UA
XLOAD15 (clrpy,vcs1) I300UA
XLOAD16 (clrny,vcs1) I300UA
XLOAD19 (d0y,vcs1) I300UA
XLOAD20 (d0yn,vcs1) I300UA
XLOAD21 (d1y,vcs1) I300UA
XLOAD22 (d1yn,vcs1) I300UA
XLOAD23 (d2y,vcs1) I300UA
XLOAD24 (d2yn,vcs1) I300UA
XLOAD25 (d3y,vcs1) I300UA
XLOAD26 (d3yn,vcs1) I300UA
XLOAD27 (d4y,vcs1) I300UA
XLOAD28 (d4yn,vcs1) I300UA
XLOAD29 (d5y,vcs1) I300UA
XLOAD30 (d5yn,vcs1) I300UA
XLOAD31 (d6y,vcs1) I300UA
XLOAD32 (d6yn,vcs1) I300UA
XLOAD33 (d7y,vcs1) I300UA
XLOAD34 (d7yn,vcs1) I300UA
XLOAD35 (mux04p,vcs3) I300UA
XLOAD36 (mux04n,vcs3) I300UA
XLOAD37 (mux26p,vcs3) I300UA
XLOAD38 (mux26n,vcs3) I300UA
XLOAD39 (mux15p,vcs3) I300UA
XLOAD40 (mux15n,vcs3) I300UA
XLOAD41 (mux37p,vcs3) I300UA
XLOAD42 (mux37n,vcs3) I300UA
XLOAD43 (mevenp,vcs3) I1MA
XLOAD44 (mevenn,vcs3) I1MA
XLOAD45 (moddp,vcs3) I1MA
XLOAD46 (moddn,vcs3) I1MA
XLOAD47 (mux0t7p,vcs2) I1MA
XLOAD47a (mux0t7p,vcs2) I550UA
XLOAD48 (mux0t7n,vcs2) I1MA
XLOAD48a (mux0t7n,vcs2) I550UA
*
*
*       CAP/DIODE CARDS
*       represents iteration #3 which went to mask shop
*       on March 9, 1990
*
*
CI15124     id0xn     0         capmod    (51298.926, 1023.500,  688.000)
CI15125     id4xn     0         capmod    (56487.125, 1317.500,  884.000)
CI15126     id4xp     0         capmod    (60491.227, 1389.500,  932.000)
CI15127     id2xp     0         capmod    (43268.898, 1013.000,  681.000)
CI15128     id2xn     0         capmod    (45653.500, 1045.000,  708.000)
CI15129     id6xn     0         capmod    (33476.000,  770.000,  519.000)
CI15130     id6xp     0         capmod    (36586.000,  820.000,  558.000)
CI15139     ckpm      0         capmod    (4090.287,    67.625,   50.750)
CI15140     cknm      0         capmod    (10637.801,  181.000,  143.000)
CI15152     mcky8p    0         capmod    (27590.400,  468.000,  346.000)
CI15153     mcky8n    0         capmod    (20314.551,  348.500,  255.000)
CI15154     mcky4p    0         capmod    (35043.824,  612.750,  442.500)
CI15155     mcky4n    0         capmod    (34609.574,  590.250,  433.000)
```

```
CI15156    mux04n   0        capmod   (41253.801,  896.000,  603.000)
CI15161    mux04p   0        capmod   (31283.602,  662.000,  447.000)
CI15163    mux26p   0        capmod   (27116.602,  572.000,  387.000)
CI15168    mux26n   0        capmod   (17146.400,  338.000,  231.000)
CI15179    mevenn   0        capmod   (4105.400,   68.000,   51.000)
CI15180    mevenp   0        capmod   (4136.156,   68.438,   51.500)
CI15181    mcky2p   0        capmod   (11017.675,  192.250,  139.500)
CI15182    mcky2n   0        capmod   (9961.825,  172.750,  126.500)
CI15184    moddp    0        capmod   (4165.850,   69.500,   52.000)
CI15185    moddn    0        capmod   (4165.850,   69.500,   52.000)
CI15191    clrxnm   0        capmod   (13281.375,  221.250,  164.500)
CI15192    clrxpm   0        capmod   (8180.575,  135.250,  101.500)
CI15199    q1xpm    0        capmod   (6007.550,  108.500,   78.000)
CI15200    mux15n   0        capmod   (41253.801,  896.000,  603.000)
CI15201    mux15p   0        capmod   (31283.602,  662.000,  447.000)
CI15202    mux37p   0        capmod   (27116.602,  572.000,  387.000)
CI15203    mux37n   0        capmod   (17146.400,  338.000,  231.000)
CI15204    q1xnm    0        capmod   (4921.475,   88.250,   64.500)
CI15210    id7xn    0        capmod   (36706.898,  823.000,  560.000)
CI15211    id7xp    0        capmod   (33596.898,  773.000,  521.000)
CI15213    id3xp    0        capmod   (43268.898, 1013.000,  681.000)
CI15214    id3xn    0        capmod   (45653.500, 1045.000,  708.000)
CI15224    q2xpm    0        capmod   (3079.775,   49.250,   38.500)
CI15227    id1xn    0        capmod   (57514.324, 1191.500,  800.000)
CI15228    id1xp    0        capmod   (60793.023, 1245.500,  836.000)
CI15229    id5xp    0        capmod   (68641.023, 1605.500, 1076.000)
CI15230    id5xn    0        capmod   (64636.926, 1533.500, 1028.000)
CI15232    q3xpm    0        capmod   (3079.775,   49.250,   38.500)
CI15238    framep   0        capmod   (6096.200,  104.000,   75.000)
CI15240    framen   0        capmod   (5556.200,  104.000,   75.000)
CK15259    mux0t7n  0        capmod   (8272.525,  138.625,  106.000)
CK15261    mux0t7p  0        capmod   (11442.976,  190.125,  146.000)
CK16794    id0xp    0        capmod   (54577.625, 1077.500,  724.000)
*
*
CQ17699    mux0t7n  0        capmod   (20792.625, 1012.500,  688.500)
CQ17700    mux0t7p  0        capmod   (20268.051,  948.750,  649.000)
CS17720    mevenn   0        capmod   (7580.525,  410.250,  282.500)
CS17721    mevenp   0        capmod   (10489.856,  607.500,  414.000)
CW17899    id0xp    0        capmod   (25669.625, 1399.500,  942.000)
CW17900    id0xn    0        capmod   (27197.824, 1453.500,  978.000)
CW17901    id4xn    0        capmod   (12292.025,  439.500,  302.000)
CW17902    id4xp    0        capmod   (11557.625,  439.500,  302.000)
CW17903    id2xp    0        capmod   (31284.350, 1554.000, 1045.000)
CW17904    id2xn    0        capmod   (30770.801, 1491.000, 1006.000)
CW17905    id6xn    0        capmod   (44382.051, 2445.000, 1639.000)
CW17906    id6xp    0        capmod   (44662.301, 2436.000, 1636.000)
CW17915    ckpm     0        capmod   (61421.602, 4036.500, 2700.000)
CW17916    cknm     0        capmod   (53096.461, 3441.375, 2306.250)
CW17926    mcky8p   0        capmod   (65274.523, 3798.750, 2556.500)
CW17927    mcky8n   0        capmod   (66883.703, 3978.000, 2670.000)
CW17928    mcky4p   0        capmod   (74803.047, 4228.500, 2843.000)
CW17929    mcky4n   0        capmod   (75824.773, 4219.500, 2844.500)
CW17930    mux04n   0        capmod   (1750.325,   68.250,   54.500)
CW17934    mux04p   0        capmod   (2680.925,  104.250,   78.500)
CW17936    mux26p   0        capmod   (3622.550,  141.000,  103.000)
CW17941    mux26n   0        capmod   (4553.150,  177.000,  127.000)
CW17948    mcky2p   0        capmod   (35639.727, 2181.750, 1466.500)
CW17949    mcky2n   0        capmod   (35311.227, 2145.750, 1442.500)
CW17950    moddp    0        capmod   (13379.675,  804.750,  545.500)
CW17951    moddn    0        capmod   (16323.350, 1005.000,  679.000)
CW17981    clrxnm   0        capmod   (46757.324, 2978.250, 2000.500)
CW17982    clrxpm   0        capmod   (47969.949, 3061.500, 2053.000)
CW17985    q1xpm    0        capmod   (5686.250,  336.000,  233.000)
```

```
CW17986     mux15n    0          capmod    (1761.350,    69.000,    55.000)
CW17987     mux15p    0          capmod    (2691.950,   105.000,    79.000)
CW17988     mux37p    0          capmod    (3622.550,   141.000,   103.000)
CW17989     mux37n    0          capmod    (4553.150,   177.000,   127.000)
CW17992     q1xnm     0          capmod    (4376.525,   233.250,   164.500)
CW17994     id7xn     0          capmod    (41917.898, 2445.000, 1642.000)
CW17995     id7xp     0          capmod    (41637.648, 2454.000, 1645.000)
CW17996     id3xp     0          capmod    (29253.051, 1566.000, 1053.000)
CW17997     id3xn     0          capmod    (28739.500, 1503.000, 1014.000)
CW18002     q2xpm     0          capmod    (8513.150,   501.000,   343.000)
CW18003     q2xnm     0          capmod    (5344.350,   313.500,   215.000)
CW18005     id1xn     0          capmod    (24770.824, 1297.500,   874.000)
CW18006     id1xp     0          capmod    (23242.625, 1243.500,   838.000)
CW18009     id5xp     0          capmod    (8425.025,   235.500,   166.000)
CW18010     id5xn     0          capmod    (9159.425,   235.500,   166.000)
CW18011     q3xpm     0          capmod    (11779.550,   705.000,   479.000)
CW18012     q3xnm     0          capmod    (7585.425,   447.750,   304.500)
CW18016     framep    0          capmod    (7229.750,   441.000,   303.000)
CW18018     framen    0          capmod    (7218.725,   440.250,   302.500)
*
*
*
*
*
*voltage card
vclrm clrm 0 table(0ps,-0.8v,300ps,-0.8v,500ps,-1.6v,100000ps,-1.6v)
vclkpm clkpm 0 sin(-1.2 0.4 &frq)
vclknm clknm 0 sin(-1.2 0.4 &frq 0 0 180)
.set &frq=2.5e+9
vmdata0 mdata0 0 -1.6v
vmdata1 mdata1 0 -0.8v
vmdata2 mdata2 0 -0.8v
vmdata3 mdata3 0 -1.6v
vmdata4 mdata4 0 -0.8v
vmdata5 mdata5 0 -1.6v
vmdata6 mdata6 0 -0.8v
vmdata7 mdata7 0 -1.6v
*
.END
dmux - typical at T=25 degrees C
.use /u3018/dbic/best1lib/advtyp.dat
*****************************************************************************
* this file represents the electrical connectivity of the Time Division
* Demultiplexer
*****************************************************************************
*
   xibg1   (vbbx1,vbby1,vcs1)    ibg ;*handles buffers
   xibg2   (vbbx2,vbby2,vcs2)    ibg ;*handles the top row of logic
   xibg3   (vbbx3,vbby3,vcs3)    ibg ;*handles the middle row of logic
   xibg4   (vbbx4,vbby4,vcs4)    ibg ;*handles the bottom row of logic
*input buffers
   xbclkdm (vbbx1,vbby1,vcs1,clkpdm,clkndm:dmckp,dmckn) ebcb
   xbdata  (vbbx1,vbby1,vcs1,datap,datan
+           :dmux0t7p,dm0t7py,dmux0t7n,dm0t7ny) ebi1n1
*top row of logic
   xdml1   (vbbx2,vbby2,vcs2,dmux0t7p,dmux0t7n,dmclk2n,dmclk2p
+           :dmoddp,dmoddn) f3dahd
   xdmff4  (vbbx2,vbby2,vcs2,dmux0t7p,dmux0t7n,dmclk2n,dmclk2p
+           :dmevenp,dmevenn) f1dahd
   xdml2   (vbbx2,vbby2,vcs2,dmevenp,dmevenn,dmclk4n,dmclk4p
+           :dmux26p,dmux26n) f3dahd
   xdmff5  (vbbx2,vbby2,vcs2,dmevenp,dmevenn,dmclk4n,dmclk4p
+           :dmux04p,dmux04n) f1dahd
   xdml3   (vbbx2,vbby2,vcs2,dmoddp,dmoddn,dmclk4n,dmclk4p
```

```
+              :dmux37p,dmux37n) f3dahd
   xdmff6   (vbbx2,vbby2,vcs2,dmoddp,dmoddn,dmclk4n,dmclk4p
+              :dmux15p,dmux15n) f1dahd
*middle row of logic
   XDMFF1   (vbbx3,vbby3,vcs3,q1xpdm,q1xndm,q1xndm,q1xpdm,sh1yp,sh1yn,clrxpdm,
+  clrxndm,clrxpdm,clrxndm,dmckp,dmckn:q1xpdm,dmclk2p,q1xndm,dmclk2n) f2dchd
   XDMFF2   (vbbx3,vbby3,vcs3,q2xpdm,q2xndm,q2xndm,q2xpdm,sh2yp,sh2yn,clrxpdm,
+  clrxndm,clrxpdm,clrxndm,dmclk2p,dmclk2n:q2xpdm,dmclk4p,q2xndm,dmclk4n) f2dchd
   XDMFF3   (vbbx3,vbby3,vcs3,q3xpdm,q3xndm,q3xndm,q3xpdm,sh4yp,sh4yn,clrxpdm,
+  clrxndm,clrxpdm,clrxndm,dmclk4p,dmclk4n:q3xpdm,dmclk8p,q3xndm,dmclk8n) f2dchd
   xbuf1dm (vbbx3,vbby3,vcs3,dmclk8p,dmclk8n:frq8p,frq8n) rcr1hd
*bottom row of logic
   xdmff7   (vbbx4,vbby4,vcs4,dmux04p,dmux04n,dmclk8p,dmclk8n,dmclk8p,dmclk8n
+              :dmux0xp,dmux0xn) f5dahd
   xdmff8   (vbbx4,vbby4,vcs4,dmux04p,dmux04n,dmclk8p,dmclk8n
+              :dmux4xp,dmux4xn) f1dahd
   xdmff9   (vbbx4,vbby4,vcs4,dmux26p,dmux26n,dmclk8p,dmclk8n,dmclk8p,dmclk8n
+              :dmux2xp,dmux2xn) f5dahd
   xdmff10  (vbbx4,vbby4,vcs4,dmux26p,dmux26n,dmclk8p,dmclk8n
+              :dmux6xp,dmux6xn) f1dahd
   xdmff11  (vbbx4,vbby4,vcs4,dmux15p,dmux15n,dmclk8p,dmclk8n,dmclk8p,dmclk8n
+              :dmux1xp,dmux1xn) f5dahd
   xdmff12  (vbbx4,vbby4,vcs4,dmux15p,dmux15n,dmclk8p,dmclk8n
+              :dmux5xp,dmux5xn) f1dahd
   xdmff13  (vbbx4,vbby4,vcs4,dmux37p,dmux37n,dmclk8p,dmclk8n,dmclk8p,dmclk8n
+              :dmux3xp,dmux3xn) f5dahd
   xdmff14  (vbbx4,vbby4,vcs4,dmux37p,dmux37n,dmclk8p,dmclk8n
+              :dmux7xp,dmux7xn) f1dahd
*output buffers
   xbdata0  (vbbx1,vbby1,vcs1,dmux0xp,dmux0xn:data0) x1mn
   xbdata4  (vbbx1,vbby1,vcs1,dmux4xp,dmux4xn:data4) x1mn
   xbdata2  (vbbx1,vbby1,vcs1,dmux2xp,dmux2xn:data2) x1mn
   xbdata6  (vbbx1,vbby1,vcs1,dmux6xp,dmux6xn:data6) x1mn
   xbdata1  (vbbx1,vbby1,vcs1,dmux1xp,dmux1xn:data1) x1mn
   xbdata3  (vbbx1,vbby1,vcs1,dmux3xp,dmux3xn:data3) x1mn
   xbdata5  (vbbx1,vbby1,vcs1,dmux5xp,dmux5xn:data5) x1mn
   xbdata7  (vbbx1,vbby1,vcs1,dmux7xp,dmux7xn:data7) x1mn
   xbfrdm   (vbbx1,vbby1,vcs1,frq8p,frq8n:frpdm,frndm) xbmb
*fanout loading
   xload3 (frpdm) ECL50
   xload4 (frndm) ECL50
   xload5 (data0) ECL50
   xload6 (data4) ECL50
   xload7 (data2) ECL50
   xload8 (data6) ECL50
   xload9 (data1) ECL50
   xload10 (data3) ECL50
   xload11 (data5) ECL50
   xload12 (data7) ECL50
*current sources
   xcur5 (dmclk2p,vcs3) I1MA
   xcur5a (dmclk2p,vcs3) I550UA
   xcur6 (dmclk2n,vcs3) I1MA
   xcur6a (dmclk2n,vcs2) I550UA
   xcur7 (dmclk4p,vcs3) I1MA
   xcur7a (dmclk4p,vcs3) I550UA
   xcur8 (dmclk4n,vcs3) I1MA
   xcur8a (dmclk4n,vcs2) I550UA
   xcur9 (dmclk8p,vcs3) I1MA
   xcur9a (dmclk8p,vcs3) I550UA
   xcur10 (dmclk8n,vcs3) I1MA
   xcur10a (dmclk8n,vcs3) I550UA
   xcur11 (dm0t7py,vcs1) I1MA
   xcur12 (dm0t7ny,vcs1) I1MA
```

```
xcur13 (dmoddp,vcs2) I1MA
xcur14 (dmoddn,vcs2) I1MA
xcur15 (dmevenp,vcs2) I1MA
xcur16 (dmevenn,vcs2) I1MA
xcur17 (dmux04p,vcs2) I300UA
xcur18 (dmux04n,vcs2) I300UA
xcur19 (dmux26p,vcs2) I300UA
xcur20 (dmux26n,vcs2) I300UA
xcur29 (dmux15p,vcs2) I300UA
xcur30 (dmux15n,vcs2) I300UA
xcur31 (dmux37p,vcs2) I300UA
xcur32 (dmux37n,vcs2) I300UA
xcur21 (dmux0xp,vcs4) I300UA
xcur22 (dmux0xn,vcs4) I300UA
xcur23 (dmux2xp,vcs4) I300UA
xcur24 (dmux2xn,vcs4) I300UA
xcur25 (dmux4xp,vcs4) I300UA
xcur26 (dmux4xn,vcs4) I300UA
xcur27 (dmux6xp,vcs4) I300UA
xcur28 (dmux6xn,vcs4) I300UA
xcur33 (dmux1xp,vcs4) I300UA
xcur34 (dmux1xn,vcs4) I300UA
xcur35 (dmux5xp,vcs4) I300UA
xcur36 (dmux5xn,vcs4) I300UA
xcur37 (dmux3xp,vcs4) I300UA
xcur38 (dmux3xn,vcs4) I300UA
xcur39 (dmux7xp,vcs4) I300UA
xcur40 (dmux7xn,vcs4) I300UA
xcur41 (frq8n,vcs3) I550UA
xcur42 (frq8p,vcs3) I550UA
*
*
*       C P/DIODE CARDS
*       this represents iteration #3 sent to the mask shop
*       on March 9, 1990
*
*
CI15084    dmclk8p  0        capmod   (40833.441,  661.500,  509.000)
CI15085    dmclk8n  0        capmod   (14006.025,  233.625,  175.000)
CI15086    dmckp    0        capmod   (7547.000,  140.000,  99.000)
CI15119    clrxndm  0        capmod   (53656.523, 1096.750,  765.000)
CI15120    clrxpdm  0        capmod   (59809.262, 1191.312,  828.000)
CI15121    dmckn    0        capmod   (7547.000,  140.000,   99.000)
CI15122    dmux0t7n 0        capmod   (3110.000,   50.000,   39.000)
CI15123    dmux0t7p 0        capmod   (3079.775,   49.250,   38.500)
CI15131    dmux04p  0        capmod   (13374.075,  230.250,  170.500)
CI15132    dmux04n  0        capmod   (17313.477,  303.250,  219.000)
CI15133    q1xpdm   0        capmod   (6220.000,  100.000,   78.000)
CI15134    dmevenn  0        capmod   (12285.976,  203.250,  152.500)
CI15136    sh1yp    0        capmod   (11381.250,  187.500,  142.000)
CI15137    sh1yn    0        capmod   (8271.250,  137.500,  103.000)
CI15138    dmux0xn  0        capmod   (3110.000,   50.000,   39.000)
CI15141    dmclk2n  0        capmod   (18234.352,  311.375,  232.500)
CI15142    dmclk2p  0        capmod   (7547.000,  140.000,   99.000)
CI15143    dmux4xp  0        capmod   (3110.000,   50.000,   39.000)
CI15144    dmevenp  0        capmod   (16722.977,  293.250,  212.500)
CI15145    dmoddn   0        capmod   (4075.175,   67.250,   50.500)
CI15150    dmoddp   0        capmod   (9330.000,  150.000,  117.000)
CI15158    dmux26p  0        capmod   (16330.051,  283.500,  206.000)
CI15159    dmux26n  0        capmod   (17325.451,  301.500,  218.000)
CI15160    q2xpdm   0        capmod   (10915.025,  191.750,  139.000)
CI15162    q2xndm   0        capmod   (3110.000,   50.000,   39.000)
CI15165    sh2yp    0        capmod   (14491.251,  237.500,  181.000)
CI15166    sh2yn    0        capmod   (14491.251,  237.500,  181.000)
```

```
CI15167    dmux2xn  0        capmod    (3110.000,    50.000,    39.000)
CI15173    dmclk4n  0        capmod    (35500.133,   614.000,   447.000)
CI15174    dmux6xp  0        capmod    (3110.000,    50.000,    39.000)
CI15175    dmclk4p  0        capmod    (20982.400,   358.000,   267.000)
CI15189    dmux15p  0        capmod    (15426.200,   254.000,   192.000)
CI15190    dmux15n  0        capmod    (14307.001,   240.000,   177.000)
CI15193    q3xpdm   0        capmod    (15869.025,   321.750,   220.000)
CI15194    q3xndm   0        capmod    (5100.800,    86.000,    63.000)
CI15195    sh4yp    0        capmod    (17750.051,   286.000,   219.000)
CI15198    sh4yn    0        capmod    (14340.313,   246.875,   187.250)
CI15209    dmux37n  0        capmod    (12711.150,   220.500,   164.000)
CI15212    dmux1xp  0        capmod    (3110.000,    50.000,    39.000)
CI15216    dmux37p  0        capmod    (7610.350,    134.500,   101.000)
CI15219    dmux5xp  0        capmod    (3110.000,    50.000,    39.000)
CI15220    frq8n    0        capmod    (12076.425,   204.750,   153.500)
CI15226    frq8p    0        capmod    (7306.075,    120.250,   91.500)
CI15235    dmux3xp  0        capmod    (3110.000,    50.000,    39.000)
CI15239    dmux7xn  0        capmod    (3110.000,    50.000,    39.000)
*
*
CQ17711    dmux15n  0        capmod    (55866.422,  2486.250,  1672.500)
CS17716    sh1yp    0        capmod    (102905.961,  6055.500,  4052.000)
CS17717    sh1yn    0        capmod    (98910.852,  6226.500,  4163.000)
CS17718    sh2yp    0        capmod    (83514.914,  4858.500,  3257.000)
CS17719    sh2yn    0        capmod    (76381.250,  4660.500,  3125.000)
CS17722    sh4yp    0        capmod    (53838.363,  2907.000,  1959.000)
CS17723    sh4yn    0        capmod    (47992.477,  2742.375,  1846.250)
CS17724    dmux37n  0        capmod    (65350.172,  3117.750,  2093.500)
CS17725    dmux1xp  0        capmod    (8111.750,   501.000,   343.000)
CS17726    dmux37p  0        capmod    (61987.016,  2946.750,  1976.500)
CS17727    dmux3xp  0        capmod    (9875.750,   621.000,   423.000)
CS17728    dmux7xn  0        capmod    (8288.150,   513.000,   351.000)
CW17844    dmclk8p  0        capmod    (92664.672,  5433.750,  3670.500)
CW17845    dmclk8n  0        capmod    (105764.227,  6246.000,  4180.500)
CW17846    dmckp    0        capmod    (31471.551,  1635.000,  1099.000)
CW17894    clrxndm  0        capmod    (88626.852,  4542.750,  3057.000)
CW17895    clrxpdm  0        capmod    (86550.516,  4365.000,  2947.500)
CW17896    dmckn    0        capmod    (30942.350,  1599.000,  1075.000)
CW17897    dmux0t7n 0        capmod    (37453.926,  1532.250,  1030.500)
CW17898    dmux0t7p 0        capmod    (37762.625,  1553.250,  1044.500)
CW17907    dmux04p  0        capmod    (63943.168,  3051.750,  2049.500)
CW17908    dmux04n  0        capmod    (61861.617,  2985.000,  2009.500)
CW17909    q1xpdm   0        capmod    (10282.525,   588.750,   404.500)
CW17910    dmux0xp  0        capmod    (12594.300,   834.000,   562.000)
CW17911    dmevenn  0        capmod    (30998.850,  1929.000,  1301.000)
CW17914    dmux0xn  0        capmod    (10493.150,   663.000,   451.000)
CW17917    dmux4xn  0        capmod    (14181.900,   942.000,   634.000)
CW17918    dmclk2n  0        capmod    (36495.762,  1928.625,  1299.250)
CW17919    dmclk2p  0        capmod    (24747.949,  1569.000,  1055.000)
CW17920    dmux4xp  0        capmod    (12080.750,   771.000,   523.000)
CW17921    dmevenp  0        capmod    (26427.750,  1659.000,  1121.000)
CW17922    dmoddn   0        capmod    (43573.625,  2813.250,  1884.500)
CW17923    dmoddp   0        capmod    (41507.926,  2630.250,  1768.500)
CW17931    dmux26p  0        capmod    (55719.066,  2427.750,  1633.500)
CW17932    dmux26n  0        capmod    (52687.977,  2391.750,  1609.500)
CW17933    q2xpdm   0        capmod    (10596.275,   546.000,   380.500)
CW17935    q2xndm   0        capmod    (9396.125,    488.250,   334.500)
CW17937    dmux2xp  0        capmod    (10742.100,   708.000,   478.000)
CW17938    dmux2xn  0        capmod    (9699.350,    609.000,   415.000)
CW17942    dmux6xn  0        capmod    (12329.700,   816.000,   550.000)
CW17943    dmclk4n  0        capmod    (49261.926,  2598.000,  1763.500)
CW17944    dmux6xp  0        capmod    (10228.550,   645.000,   439.000)
CW17945    dmclk4p  0        capmod    (37652.125,  2216.250,  1498.500)
CW17980    dmux15p  0        capmod    (61147.566,  2765.250,  1861.500)
```

```
    CW17983     q3xpdm   0          capmod   (5034.825,  264.000,  189.500)
    CW17984     q3xndm   0          capmod   (6377.300,  337.500,  234.000)
    CW17993     dmux1xn  0          capmod   (9066.300,  594.000,  402.000)
    CW17998     dmux5xp  0          capmod   (7758.950,  477.000,  327.000)
    CW17999     dmux5xn  0          capmod   (8184.300,  534.000,  362.000)
    CW18001     frq8n    0          capmod   (39410.699, 2235.000, 1505.000)
    CW18004     frq8p    0          capmod   (36185.648, 2050.500, 1379.000)
    CW18013     dmux3xn  0          capmod   (10918.500,  720.000,  486.000)
    CW18017     dmux7xp  0          capmod   (9330.900,  612.000,  414.000)
*
*
*
*
*
*voltage card
vsh1yn sh1yn 0 dc -1.2v
vsh1yp sh1yp 0 dc -0.8v
vsh2yn sh2yn 0 dc -1.2v
vsh2yp sh2yp 0 dc -0.8v
vsh4yn sh4yn 0 dc -1.2v
vsh4yp sh4yp 0 dc -0.8v
vclrxpdm clrxpdm 0 table(0ps,-0.8v,300ps,-0.8v,500ps,-1.2v,100000ps,-1.2v)
vclrxndm clrxndm 0 table(0ps,-1.2v,300ps,-1.2v,500ps,-0.8v,100000ps,-0.8v)
vclkpdm clkpdm 0 sin(-1.2 0.4 &frq)
vclkndm clkndm 0 sin(-1.2 0.4 &frq 0 0 180)
.set &frq=2.5e+9
vdatap datap 0 table(0ps,-0.8v,3300ps,-0.8v,3350ps,-1.6v,3700ps,-1.6v,3750ps,
+                    -0.8v,4100ps,-0.8v,4150ps,-1.6v,4500ps,-1.6v,4550ps,
+                    -0.8v,5300ps,-0.8v,5350ps,-1.6v,6100ps,-1.6v,6150ps,
+                    -0.8v,6900ps,-0.8v,6950ps,-1.6v,7300ps,-1.6v,7350ps,
+                    -0.8v,7700ps,-0.8v,7750ps,-1.6v,8500ps,-1.6v,8550ps,
+                    -0.8v,8900ps,-0.8v,8950ps,-1.6v,9700ps,-1.6v,9750ps,
+                    -0.8v,12000ps,-0.8v)
vdatan datan 0 table(0ps,-1.6v,3300ps,-1.6v,3350ps,-0.8v,3700ps,-0.8v,3750ps,
+                    -1.6v,4100ps,-1.6v,4150ps,-0.8v,4500ps,-0.8v,4550ps,
+                    -1.6v,5300ps,-1.6v,5350ps,-0.8v,6100ps,-0.8v,6150ps,
+                    -1.6v,6900ps,-1.6v,6950ps,-0.8v,7300ps,-0.8v,7350ps,
+                    -1.6v,7700ps,-1.6v,7750ps,-0.8v,8500ps,-0.8v,8550ps,
+                    -1.6v,8900ps,-1.6v,8950ps,-0.8v,9700ps,-0.8v,9750ps,
+                    -1.6v,12000ps,-1.6v)
.END
*The following lines represent the device models used in the
*simulations
**********************************************************************
* references and supplies                                            *
**********************************************************************
*
vee vee 0 dc -5.2
rvee vee 0 10g
vgnd gnd 0 dc 0
rgnd gnd 0 10g
vcc vcc 0 dc 5.0
rvcc vcc 0 10g
*
xibg vbbx vbby vcs ibg
rvbbx vbbx 0 10g
rvbby vbby 0 10g
rvcs vcs 0 10g
*
.temp 25
**********************************************************************
*   transistor models typical - 9/11/89                              *
**********************************************************************
*
```

```
* Model for ring-dot pnp (Preliminary)
*
.model pnpv pnp (rbx=500 rbi=1000 rc=80
+ re=0 ccs=0 cbs=0.01pf is=9.5e-18
+ ik=2e-4 vao=0 tfo=1.5e-09 cje=0.045pf
+ bf=35 i2=5.86e-15 ne=2
+ pe=0.75 me=0.333 be=0.10 i3=1.01e-20
+ i4=0 nc=2 ikr=5e-05 vbo=0 tro=1.5e-09
+ cjc=0.01pf pc=0.75 mc=0.333 bc=0.10
+ td=0 ea=1.24 dea=8e-02 to=25
*
*
* Model for elxr 1.5x3 um emitter Bf=125
* 4 terminal model collector to bottom
 .MODEL me1xr  NPN ( RBX = 40    RBI =  625    RC  =  150
 +  RE  =  32     IS  =  6.25E-18
 +  I1  =  5.0E-20 I2  = 160E-18 NE  = 1.750 IK  =  9E-3
 +  VBO =4  TFO = 13.26E-12  CJE = 12E-15  PE  =  1
 +  ME  =  0.688  BE  =  1.000E-01 I3  =  1E-20 I4 = 0.
 +  NC  =  2.000E+00 IKR =  1E-4 VAO =  30 TRO =  0.
 +  CJC =  7.5E-15    PC  =  0.818 MC  =  0.234 BC  =  1.000E-01
 +  EA  =  1.206E+00 DEA =  0.062 TO  =  2.500E+01
* +  BVBC = 15    ALC1=  2.000E+00   ALC2=  0    ALTC=  7.500E-01
* +  BVBE=3        ALE1=  2.000E+00 ALE2=  0   ALTE=  7.500E-01
 +  KFN =  0    AFN =  1.000E+00   BFN =  1.000E+00
 +  IKS =1E-5
 +  ISS = 2.35E-18 ISC = 1E-16
 +  CJS =  32E-15    PS  =  0.836   MS  =  0.5 BS  =  1.000E-01)
*
* Model for e8xr 8(1.5x3) emitters
* Emitter stripes in row with collector to bottom
* 4 terminal model
 .MODEL me8xr  NPN ( RBX = 6    RBI =  82    RC  =  17
 +  RE  =  6     IS  =  50E-18
 +  I1  =  40E-20 I2  = 128E-17 NE  = 1.750 IK  =  72E-3
 +  VBO =4  TFO = 13.26E-12  CJE = 96E-15  PE  =  1
 +  ME  =  0.688  BE  =  1.000E-01 I3  =  8E-20 I4 = 0.
 +  NC  =  2.000E+00 IKR =  8E-4 VAO =  30 TRO =  0.
 +  CJC =  40E-15    PC  =  0.818 MC  =  0.234 BC  =  1.000E-01
 +  EA  =  1.206E+00 DEA =  0.062 TO  =  2.500E+01
* +  BVBC = 15    ALC1=  2.000E+00   ALC2=  0    ALTC=  7.500E-01
* +  BVBE=3        ALE1=  2.000E+00 ALE2=  0   ALTE=  7.500E-01
 +  KFN =  0    AFN =  1.000E+00   BFN =  1.000E+00
 +  IKS =8E-5
 +  ISS = 18.8E-18 ISC = 8E-16
 +  CJS =  66.4E-15    PS  =  0.836   MS  =  0.5 BS  =  1.000E-01)
*
* Model for e13xr 8(1.5x5) emitters Bf=125
* Emitter stripes in row with collector to bottom
* 4 terminal model
 .MODEL me13xr  NPN ( RBX = 4    RBI =  47    RC  =  17
 +  RE  =  5     IS  =  81.2E-18
 +  I1  =  65E-20 I2  = 208E-17 NE  = 1.750 IK  =  117E-3
 +  VBO =4  TFO = 13.26E-12  CJE = 156E-15  PE  =  1
 +  ME  =  0.688  BE  =  1.000E-01 I3  =  13E-20 I4 = 0.
 +  NC  =  2.000E+00 IKR =  13E-4 VAO =  30 TRO =  0.
 +  CJC =  66E-15    PC  =  0.818 MC  =  0.234 BC  =  1.000E-01
 +  EA  =  1.206E+00 DEA =  0.062 TO  =  2.500E+01
* +  BVBC = 15    ALC1=  2.000E+00   ALC2=  0    ALTC=  7.500E-01
* +  BVBE=3        ALE1=  2.000E+00 ALE2=  0   ALTE=  7.500E-01
 +  KFN =  0    AFN =  1.000E+00   BFN =  1.000E+00
 +  IKS =13E-5
 +  ISS = 30.55E-18 ISC = 13E-16
 +  CJS =  71.2E-15    PS  =  0.836   MS  =  0.5 BS  =  1.000E-01)
```

```
*
* Model for e53x two 16(1.5x5) emitter transistors Bf=125
* 4 terminal model two collectors in center
 .MODEL me53x  NPN ( RBX = 2    RBI =  13    RC  =  5
 +  RE  =  2     IS  =  333E-18
 +  I1  =  266E-20 I2  = 853E-17 NE  = 1.750 IK  =  480E-3
 +  VBO =4  TFO = 13.26E-12  CJE = 636E-15  PE  =  1
 +  ME  =  0.6168  BE  =  1.000E-01 I3  =  53.3E-20 I4 = 0.
 +  NC  =  2.000E+00 IKR =  53.3E-4 VAO =  30 TRO =  0.
 +  CJC =  260E-15    PC  =  0.818 MC  =  0.234 BC  =  1.000E-01
 +  EA  =  1.206E+00 DEA =  0.062 TO  =  2.500E+01
* +  BVBC = 15    ALC1=  2.000E+00   ALC2=  0    ALTC=  7.500E-01
* +  BVBE=3        ALE1=  2.000E+00 ALE2=  0   ALTE=  7.500E-01
 +  KFN =  0    AFN =  1.000E+00   BFN =  1.000E+00
 +  IKS =53.3E-5
 +  ISS = 125E-18 ISC = 53.3E-16
 +  CJS =  212E-15    PS  =  0.836   MS  =  0.5 BS  =  1.000E-01)
*****************************************************************
*    resistor models                                            *
*****************************************************************
*
* Model p2w4 - poly 2 resistor, 4 um wide
.model p2w4 r tc1=-900 tc2=0.5 to=25 scal=1.0
*
* Model p2w6 - poly 2 resistor, 6 um wide
.model p2w6 r tc1=-900 tc2=0.5 to=25 scal=1.0
*
* Model p2w7 - poly 2 resistor, 7 um wide
.model p2w7 r tc1=-900 tc2=0.5 to=25 scal=1.0
*
* Model p2w10 - poly 2 resistor, 10 um wide
.model p2w10 r tc1=-900 tc2=0.5 to=25 scal=1.0
*
* Model p2w12 - poly 2 resistor, 12 um wide
.model p2w12 r tc1=-900 tc2=0.5 to=25 scal=1.0
*
* Model p2w90 - poly 2 resistor, 90 um wide
.model p2w90 r tc1=-900 tc2=0.5 to=25 scal=1.0
*
*****************************************************************
*    capacitor models                                           *
*****************************************************************
*
* Model p2cap - p2 resistor parasitic capacitor
.model p2cap c sc1=1.0
*
* Model capmod - metal and poly routing capacitor
.model capmod c sc1=1.0e-18
*
*****************************************************************
*    diode models                                               *
*****************************************************************
*
.model m3x sbd (rs=100 i1=0.3e-16 cjo=46.4e-15)
*
.model mschky1 sbd (rs=24 i1=0 cjo=0.150pf i2=7.83e-14
+  n=1.028 ea=0.965
*
.model mschkyg1 sbd (rs=24 i1=0 cjo=0.150pf i2=7.83e-14
+  n=1.028 ea=0.965
*
.model mdcap d (cjo=0.2pf)
*****************************************************************
* The following lines represent the device models for the
```

```
* simulation
*****************************************************************************
*    transistor subcircuits                                                 *
*****************************************************************************
*
.subckt e1xr c b e
.global vee
b1 c b e vee me1xr
.finis
*
.subckt e8xr c b e
.global vee
b1 c b e vee me8xr
.finis
*
.subckt e13xr c b e
.global vee
b1 c b e vee me13xr
.finis
*
.subckt e53x c b e
.global vee
b1 c b e vee me53x
.finis
*
.subckt rd175 c b e
q1 c b e pnpv
.finis
*
*****************************************************************************
* resistor subcircuits                                                      *
*****************************************************************************
*
.subckt p0p6k7 a b
r1 a b p2w7 600
ca a 0 p2cap 4.6e-15
cb b 0 p2cap 4.6e-15
.finis
*
.subckt p1k4 a b
r1 a b p2w4 1000
ca a 0 p2cap 2.77e-15
cb b 0 p2cap 2.77e-15
.finis
*
.SUBCKT p1p51k4 (a b)
r1 a b p2w4 1506
ca a 0 p2cap 4.0e-15
cb b 0 p2cap 4.0e-15
.finis
*
.subckt p2k4s a b
r1 a b p2w4 1975
ca a 0 p2cap 7.0e-15
cb b 0 p2cap 7.0e-15
.finis
*
.subckt p0p5k5 a b
r1 a b p2w4 482
ca a 0 p2cap 3.5e-15
cb b 0 p2cap 3.5e-15
.finis
*
.subckt p0p09k24 a b
```

```
r1 a b p2w4 92
ca a 0 p2cap 12e-15
cb b 0 p2cap 12e-15
.finis
*
.subckt p0p35k12 a b
r1 a b p2w4 350
ca a 0 p2cap 8.1e-15
cb b 0 p2cap 8.1e-15
.finis
*
*****************************************************************************
* diode subcircuits                                                          *
*****************************************************************************
*
.subckt d256g a b
d1 a b mschkyg1
.finis
*
.subckt c0p2 a b
x1 a a b e1xr scale=16.67
.finis
*
*****************************************************************************
* internal current sources for output loads
*****************************************************************************
*
.subckt i1ma a,vcs ;* this represents the gred clump I1MA.X
.global vee
x1  a  vcs  s1  e1xr
xr1  s1  vee  p0p6k7
xr2  s1  vee  p0p6k7
CI32 s1  0    capmod   (4212.700,   51.500,   43.500)
c1 s1 0 0.1pf
.finis
*
.subckt i550ua a,vcs ;* this represents the gred clump I550UA.X
.global  vee
x1  a  vcs  s1  e1xr
xr1  s1  vee  p0p6k7
CI32 s1  0    capmod   (4212.700,   51.500,   43.500)
c1 s1 0 0.1pf
.finis
*
.subckt i300ua a,vcs ;* this represents the gred clump I300UA.X
.global  vee
x1  a  vcs  s1  e1xr
xr1  s1  s2  p0p6k7
xr2  s2  vee  p0p6k7
CI32 s1  0    capmod   (4212.700,   51.500,   43.500)
c1 s1 0 0.1pf
.finis
*
*****************************************************************************
* load circuits for buffer characterization                                  *
*****************************************************************************
* SOURCE 1MA-- SINK 24 MA
.SUBCKT ECL50 6
xibg (vbbx,vbby,vcs) ibg
xbuf (vbbx,vbby,vcs,6:1,1,2,2) ei1n1
r12 1 0 8.8k
r13 2 0 8.8k
RL1 6 a 50
CL 6 0 1.2PF
```

```
va a 0 dc -2
.FINIS
*
*****************************************************************
* subcircuit for band-gap regulator - wrg 1/4/90                *
*****************************************************************
*
 .SUBCKT ibg (vbbx,vbby,vcs)
*
 .GLOBAL vee,gnd
*
*        SUBCIRCUIT CALLS
*
   X1        (vee,s8,s3)  rd175;* rd175#1
   X2        (vee,s5)  p1k4;* p1k4#1
   X3        (vee,s5)  p1k4;* p1k4#2
   X4        (vee,s5)  p1k4;* p1k4#3
   X5        (vee,s5)  p1k4;* p1k4#4
   X6        (vee,s5)  p1k4;* p1k4#5
   X7        (vee,s5)  p1k4;* p1k4#6
   X8        (s1,vbby)  p1k4;* p1k4#7
   X9        (s3,s9)  p1k4;* p1k4#8
   X10       (s9,gnd)  p1k4;* p1k4#9
   X11       (s2,gnd)  p1k4;* p1k4#10
   X12       (gnd,s2,vbbx)  e8xr;* e8xr#1
   X13       (gnd,vbbx,vbby)  e8xr;* e8xr#2
   X14       (s2,s3,s7)  e8xr;* e8xr#3
   X15       (vbbx,s3,vcs)  e8xr;* e8xr#4
   X16       (vee,s1)  p1k4;* p1k4#11
   X17       (s8,s3)  p1k4;* p1k4#12
   X18       (s8,s3)  p1k4;* p1k4#13
   X19       (s6,s7)  p1k4;* p1k4#14
   X21       (s4,vcs)  p1k4;* p1k4#16
   X22       (s4,vcs)  p1k4;* p1k4#17
   X23       (s4,vcs)  p1k4;* p1k4#18
   X24       (s6,s8)  c0p2;* c0p2#1
   X25       (s8,s6,vee)  e8xr;* e8xr#5
   X26       (s6,s4,s5)  e8xr;* e8xr#6
   X27       (s4,s4,vee)  e8xr;* e8xr#7
   X28       (vee,s5)  p1k4;* p1k4#19
   X29       (vee,s5)  p1k4;* p1k4#20
   X30       (vee,s5)  p1k4;* p1k4#21
   X31       (vee,s5)  p1k4;* p1k4#22
   X32       (vee,s5)  p1k4;* p1k4#23
   X33       (vee,s5)  p1k4;* p1k4#24
*
*        CAP/DIODE CARDS
*
*
*
   CI1          s1          0          capmod    (1687.200,    24.000,    20.000)
   CI2          s3          0          capmod    (23246.131,   374.500,   190.500)
   CI3          s8          0          capmod    (10772.487,   167.875,   115.000)
   CI4          s6          0          capmod    (29380.197,   485.875,   180.000)
   CK5          s9          0          capmod    (1675.800,    28.500,    32.500)
   CK6          s5          0          capmod    (14651.399,   248.000,   132.000)
   CK7          s4          0          capmod    (35524.398,   627.375,   219.500)
   CK8          s7          0          capmod    (4552.700,    79.000,    47.500)
   CK9          s2          0          capmod    (7367.925,    119.438,   102.750)
*
*
   CU10         s2          0          capmod    (17182.801,   296.000,   155.750)
   CW11         s3          0          capmod    (26134.207,   583.125,   388.750)
   CW12         s8          0          capmod    (12673.475,   277.500,   188.000)
   CW13         s6          0          capmod    (4175.375,    77.500,    46.500)
```

```
*
 .FINIS
*
*********************************************************************************
* The following lines represent the subcircuit descriptions
* for the internal and external building blocks
*********************************************************************************
 .SUBCKT ebcb (vbbx,vbby,vcs,in,ii:oy,oyn)
*
 .GLOBAL vee,gnd
*
*       SUBCIRCUIT CALLS
*
   X1        (s9,vcs,s18)   e1xr;* b/q19
   X2        (s9,vcs,s17)   e1xr;* b/q20
   X3        (oy,vcs,s12)   e1xr;* b/q21
   X4        (vee,s18)   p0p5k5;* b/r21
   X5        (vee,s18)   p0p5k5;* b/r22
   X6        (vee,s17)   p0p5k5;* b/r23
   X7        (vee,s17)   p0p5k5;* b/r24
   X8        (vee,s12)   p0p5k5;* b/r25
   X9        (vee,s12)   p0p5k5;* b/r26
   X10       (s7,gnd)   p0p35k12;* b/res4b#1/r18
   X11       (s7,gnd)   p0p35k12;* b/res4b#1/r17
   X12       (s5b,vee)   p0p09k24;* b/res3b#1/r19
   X13       (s5b,s5)   p0p09k24;* b/res3b#1/r20
   X20       (vee,s11)   p0p5k5;* b/eclin#1/res1b#1/r1
   X21       (vee,s11)   p0p5k5;* b/eclin#1/res1b#1/r2
   X31       (gnd,s9,oy)   e1xr;* b/eclin#1/q5
   X32       (gnd,s9,oy)   e1xr;* b/eclin#1/q4
   X35       (oy,vcs,s11)   e1xr;* b/eclin#1/q1
   X46       (s7,ii,s2)   e13xr;* b/q16
   X47       (vee,ii)   d256g;* b/d2
   X48       (ii,gnd)   d256g;* b/d1
   X49       (gnd,s7,s9)   e53x;* b/q15
   X50       (s10,vcs,s16)   e1xr;* a/q19
   X51       (s10,vcs,s15)   e1xr;* a/q20
   X52       (oyn,vcs,s14)   e1xr;* a/q21
   X53       (vee,s16)   p0p5k5;* a/r21
   X54       (vee,s16)   p0p5k5;* a/r22
   X55       (vee,s15)   p0p5k5;* a/r23
   X56       (vee,s15)   p0p5k5;* a/r24
   X57       (vee,s14)   p0p5k5;* a/r25
   X58       (vee,s14)   p0p5k5;* a/r26
   X59       (s8,gnd)   p0p35k12;* a/res4b#1/r18
   X60       (s8,gnd)   p0p35k12;* a/res4b#1/r17
   X61       (s5a,s5)   p0p09k24;* a/res3b#1/r19
   X62       (s5a,vee)   p0p09k24;* a/res3b#1/r20
   X69       (vee,s13)   p0p5k5;* a/eclin#1/res1b#1/r1
   X70       (vee,s13)   p0p5k5;* a/eclin#1/res1b#1/r2
   X80       (gnd,s10,oyn)   e1xr;* a/eclin#1/q5
   X81       (gnd,s10,oyn)   e1xr;* a/eclin#1/q4
   X84       (oyn,vcs,s13)   e1xr;* a/eclin#1/q1
   X93       (s2,vcs,s5)   e8xr;* a/q18
   X95       (s8,in,s2)   e13xr;* a/q16
   X96       (vee,in)   d256g;* a/d2
   X97       (in,gnd)   d256g;* a/d1
   X98       (gnd,s8,s10)   e53x;* a/q15
*
*       CAP/DIODE CARDS
*
*
   CE1          s5          0          capmod    (28067.469,   325.000,   170.500)
   CG2          s5b         0          capmod    (60556.801,   456.000,    86.000)
```

```
CG3         s5a         0          capmod    (60556.801,   456.000,    86.000)
CI4         s9          0          capmod    (34191.086,   538.625,   336.000)
CI5         s8          0          capmod    (29268.744,   481.750,   321.500)
CI6         oyn         0          capmod    (12835.534,   231.000,   139.000)
CI7         s13         0          capmod    (2437.600,    39.500,    33.500)
CK8         ii          0          capmod    (54847.250,   871.750,   424.500)
CK9         oy          0          capmod    (12835.534,   231.000,   139.000)
CK10        s11         0          capmod    (2437.600,    39.500,    33.500)
CK11        s12         0          capmod    (2322.600,    39.500,    33.500)
CK12        s17         0          capmod    (2322.600,    39.500,    33.500)
CK13        s7          0          capmod    (29268.744,   481.750,   321.500)
CK14        s18         0          capmod    (2322.600,    39.500,    33.500)
CK15        s2          0          capmod    (25293.449,   437.750,   188.500)
CK16        s10         0          capmod    (34191.086,   538.625,   336.000)
CK17        in          0          capmod    (54847.250,   871.750,   424.500)
CK18        s16         0          capmod    (2322.600,    39.500,    33.500)
CK19        s15         0          capmod    (2322.600,    39.500,    33.500)
CK20        s14         0          capmod    (2322.600,    39.500,    33.500)
*
*
   CQ21        oyn         0          capmod    (6576.450,   140.625,    98.250)
   CU22        ii          0          capmod    (27748.557,   698.750,   136.000)
   CU23        s9          0          capmod    (44661.273,   893.750,   332.500)
   CU24        oy          0          capmod    (6576.450,   140.625,    98.250)
   CU25        s10         0          capmod    (44661.273,   893.750,   332.500)
   CU26        in          0          capmod    (27748.557,   698.750,   136.000)
*
*
*
*
 .FINIS
*
 .SUBCKT ebiln1 (vbbx,vbby,vcs,in1,ii1:ox,oy,oxn,oyn)
*
 .GLOBAL vee,gnd
*
*       SUBCIRCUIT CALLS
*
   X47       (vee,in1)   d256g;* b/d2
   X48       (in1,gnd)   d256g;* b/d1
   X63       (s7,gnd)   p2k4s;* a/eclin#1/res2db#1/r16
   X64       (s7,gnd)   p2k4s;* a/eclin#1/res2db#1/r15
   X65       (s8,gnd)   p2k4s;* a/eclin#1/res2db#1/r14
   X66       (s8,gnd)   p2k4s;* a/eclin#1/res2db#1/r13
   X76       (vee,s5)   p1p51k4;* a/eclin#1/res1b#1/r10
   X77       (vee,s5)   p1p51k4;* a/eclin#1/res1b#1/r9
   X81       (s2,vcs,s5)   elxr;* a/eclin#1/q4
   X85       (s7,ii1,s2)   elxr;* a/eclin#1/q9
   X86       (ox,ox,oy)   elxr;* a/eclin#1/q10
   X88       (gnd,s7,ox)   elxr;* a/eclin#1/q14
   X90       (gnd,s8,oxn)   elxr;* a/eclin#1/q13
   X91       (oxn,oxn,oyn)   elxr;* a/eclin#1/q7
   X92       (s8,in1,s2)   elxr;* a/eclin#1/q8
   X96       (vee,ii1)   d256g;* a/d2
   X97       (ii1,gnd)   d256g;* a/d1
*
*       CAP/DIODE CARDS
*
*
*
   CG1         s5          0          capmod    (4077.800,    36.000,    32.000)
   CI2         oyn         0          capmod    (2915.113,    42.875,    38.500)
   CI3         oy          0          capmod    (2915.113,    42.875,    38.500)
   CK4         in1         0          capmod    (39620.648,   651.812,   257.250)
   CK5         oxn         0          capmod    (5136.938,    76.875,    60.000)
```

```
    CK6         ii1         0         capmod    (29850.988,   492.125,   166.000)
    CK7         s8          0         capmod    (8550.199,    132.750,   124.750)
    CK8         s2          0         capmod    (2839.900,     50.500,    55.500)
    CK9         s7          0         capmod    (7906.200,    132.750,   124.750)
    CK10        ox          0         capmod    (5136.938,     76.875,    60.000)
*
*
    CS11        oxn         0         capmod    (18214.938,   442.125,   297.750)
    CU12        in1         0         capmod    (61217.145, 2646.500,   550.250)
    CU13        oyn         0         capmod    (10336.200,   264.750,   179.500)
    CU14        ii1         0         capmod    (58581.695, 1376.250,   271.500)
    CU15        ox          0         capmod    (17892.588,   442.125,   297.750)
    CU16        oy          0         capmod    (10336.200,   264.750,   179.500)
*
*
*
*
  .FINIS
*
  .SUBCKT ei1n1 (vbbx,vbby,vcs,in1:ox,oy,oxn,oyn)
*
  .GLOBAL vee,gnd
*
*         SUBCIRCUIT CALLS
*
    X14         (s7,gnd)   p2k4s;* a/eclin#1/res2db#1/r16
    X15         (s7,gnd)   p2k4s;* a/eclin#1/res2db#1/r15
    X16         (s8,gnd)   p2k4s;* a/eclin#1/res2db#1/r14
    X17         (s8,gnd)   p2k4s;* a/eclin#1/res2db#1/r13
    X24         (vee,s4)   p1p51k4;* a/eclin#1/res1b#1/r4
    X25         (vee,s6)   p1p51k4;* a/eclin#1/res1b#1/r12
    X26         (vee,s6)   p1p51k4;* a/eclin#1/res1b#1/r11
    X27         (vee,s5)   p1p51k4;* a/eclin#1/res1b#1/r10
    X28         (vee,s5)   p1p51k4;* a/eclin#1/res1b#1/r9
    X29         (vee,s4)   p1p51k4;* a/eclin#1/res1b#1/r3
    X30         (s3,vcs,s6)   e1xr;* a/eclin#1/q6
    X32         (s2,vcs,s5)   e1xr;* a/eclin#1/q4
    X33         (s1,vcs,s4)   e1xr;* a/eclin#1/q3
    X36         (s7,s3,s2)   e1xr;* a/eclin#1/q9
    X37         (ox,ox,oy)   e1xr;* a/eclin#1/q10
    X38         (gnd,vbbx,s3)   e1xr;* a/eclin#1/q12
    X39         (gnd,s7,ox)   e1xr;* a/eclin#1/q14
    X40         (gnd,in1,s1)   e1xr;* a/eclin#1/q11
    X41         (gnd,s8,oxn)   e1xr;* a/eclin#1/q13
    X42         (oxn,oxn,oyn)   e1xr;* a/eclin#1/q7
    X43         (s8,s1,s2)   e1xr;* a/eclin#1/q8
    X47         (vee,in1)   d256g;* a/d2
    X48         (in1,gnd)   d256g;* a/d1
*
*         CAP/DIODE CARDS
*
*
    CG1         s4          0         capmod    (8030.050,    83.500,    79.500)
    CG2         s5          0         capmod    (4077.800,    36.000,    32.000)
    CI3         oyn         0         capmod    (2915.113,    42.875,    38.500)
    CI4         s6          0         capmod    (6087.650,    55.500,    51.500)
    CI5         s3          0         capmod    (8083.550,   141.000,   145.000)
    CI6         oy          0         capmod    (2915.113,    42.875,    38.500)
    CK7         in1         0         capmod    (39687.914,   620.312,   239.250)
    CK8         oxn         0         capmod    (5136.938,    76.875,    60.000)
    CK9         s1          0         capmod    (8083.550,   141.000,   145.000)
    CK10        s8          0         capmod    (7676.200,   132.750,   124.750)
    CK11        s2          0         capmod    (2839.900,    50.500,    55.500)
    CK12        s7          0         capmod    (7676.200,   132.750,   124.750)
```

```
   CK13        ox        0         capmod    (5136.938,    76.875,    60.000)
*
*
   CU14        in1       0         capmod    (27748.557,   698.750,   136.000)
   CU15        oyn       0         capmod    (10336.200,   264.750,   179.500)
   CU16        oxn       0         capmod    (18298.699,   442.125,   297.750)
   CU17        ox        0         capmod    (18298.699,   442.125,   297.750)
   CU18        oy        0         capmod    (10336.200,   264.750,   179.500)
*
 .FINIS
*
 .SUBCKT fldahd (vbbx,vbby,vcs,dx,dxn,cky,ckyn:qx,qxn)
*
 .GLOBAL vee,gnd
*
*        SUBCIRCUIT CALLS
*
   X1        (s15b,gnd)  p2k4s;* b/scdiff#1/res2d#1/r22
   X2        (s15b,gnd)  p2k4s;* b/scdiff#1/res2d#1/r21
   X5        (s16b,gnd)  p2k4s;* b/scdiff#1/res2d#1/r18
   X6        (s16b,gnd)  p2k4s;* b/scdiff#1/res2d#1/r17
   X11       (vee,s19b)  p1p51k4;* b/scdiff#1/res1a#1/r8
   X13       (vee,s19b)  p1p51k4;* b/scdiff#1/res1a#1/r7
   X16       (vee,s14b)  p1p51k4;* b/scdiff#1/res1a#1/r16
   X17       (vee,s20b)  p1p51k4;* b/scdiff#1/res1a#1/r14
   X18       (vee,s14b)  p1p51k4;* b/scdiff#1/res1a#1/r15
   X19       (vee,s20b)  p1p51k4;* b/scdiff#1/res1a#1/r13
   X23       (s17b,vcs,s19b)  e1xr;* b/scdiff#1/q7
   X24       (s13b,vcs,s14b)  e1xr;* b/scdiff#1/q6
   X25       (s18b,vcs,s20b)  e1xr;* b/scdiff#1/q5
   X30       (s12b,ckyn,s13b)  e1xr;* b/scdiff#1/q11
   X32       (gnd,s15b,s17b)  e1xr;* b/scdiff#1/q19
   X33       (gnd,s15b,qx)  e1xr;* b/scdiff#1/q21
   X34       (s16b,s17b,s12b)  e1xr;* b/scdiff#1/q17
   X35       (s15b,s18b,s12b)  e1xr;* b/scdiff#1/q16
   X36       (gnd,s16b,s18b)  e1xr;* b/scdiff#1/q18
   X37       (gnd,s16b,qxn)  e1xr;* b/scdiff#1/q20
   X39       (s15b,s7b,s11b)  e1xr;* b/scdiff#1/q14
   X40       (s16b,s8b,s11b)  e1xr;* b/scdiff#1/q15
   X41       (s11b,cky,s13b)  e1xr;* b/scdiff#1/q10
   X44       (s6b,gnd)  p2k4s;* a/scdiff#1/res2d#1/r22
   X45       (s6b,gnd)  p2k4s;* a/scdiff#1/res2d#1/r21
   X48       (s5b,gnd)  p2k4s;* a/scdiff#1/res2d#1/r18
   X49       (s5b,gnd)  p2k4s;* a/scdiff#1/res2d#1/r17
   X54       (vee,s10b)  p1p51k4;* a/scdiff#1/res1a#1/r8
   X56       (vee,s10b)  p1p51k4;* a/scdiff#1/res1a#1/r7
   X59       (vee,s9b)  p1p51k4;* a/scdiff#1/res1a#1/r16
   X60       (vee,s4b)  p1p51k4;* a/scdiff#1/res1a#1/r14
   X61       (vee,s9b)  p1p51k4;* a/scdiff#1/res1a#1/r15
   X62       (vee,s4b)  p1p51k4;* a/scdiff#1/res1a#1/r13
   X66       (s8b,vcs,s10b)  e1xr;* a/scdiff#1/q7
   X67       (s7b,vcs,s9b)  e1xr;* a/scdiff#1/q6
   X68       (s3b,vcs,s4b)  e1xr;* a/scdiff#1/q5
   X73       (s1b,cky,s3b)  e1xr;* a/scdiff#1/q11
   X76       (gnd,s6b,s8b)  e1xr;* a/scdiff#1/q21
   X78       (s6b,s7b,s1b)  e1xr;* a/scdiff#1/q16
   X80       (gnd,s5b,s7b)  e1xr;* a/scdiff#1/q20
   X81       (s5b,dx,s2b)  e1xr;* a/scdiff#1/q13
   X82       (s6b,dxn,s2b)  e1xr;* a/scdiff#1/q14
   X83       (s5b,s8b,s1b)  e1xr;* a/scdiff#1/q15
   X84       (s2b,ckyn,s3b)  e1xr;* a/scdiff#1/q10
*
*        CAP/DIODE CARDS
*
```

```
*
CG1         s19b        0          capmod    (8412.800,    83.500,    79.500)
CG2         s14b        0          capmod    (4211.375,    35.000,    31.000)
CG3         s20b        0          capmod    (4676.425,    41.000,    37.000)
CG4         s10b        0          capmod    (7943.800,    83.500,    79.500)
CG5         s9b         0          capmod    (3945.000,    35.000,    31.000)
CG6         s4b         0          capmod    (4676.425,    41.000,    37.000)
CI7         qx          0          capmod    (2734.887,    41.500,    45.000)
CI8         s17b        0          capmod    (14115.662,   229.500,   233.500)
CI9         ckyn        0          capmod    (13532.676,   292.250,   296.500)
CI10        s12b        0          capmod    (2707.800,    51.000,    55.000)
CI11        cky         0          capmod    (9655.925,    194.750,   197.000)
CI12        s13b        0          capmod    (2677.300,    53.500,    57.500)
CI13        s11b        0          capmod    (2431.675,    53.500,    57.500)
CI14        s8b         0          capmod    (13796.212,   222.375,   218.500)
CI15        qxn         0          capmod    (2733.450,    41.500,    45.000)
CI16        s1b         0          capmod    (4780.538,    85.125,    76.000)
CI17        s3b         0          capmod    (2336.050,    53.500,    57.500)
CI18        s5b         0          capmod    (12759.763,   212.125,   191.500)
CI19        s2b         0          capmod    (3973.100,    77.000,    80.500)
CI20        dxn         0          capmod    (1455.375,    26.250,    28.000)
CI21        dx          0          capmod    (1696.887,    27.125,    22.500)
CK22        s15b        0          capmod    (16161.275,   263.625,   222.500)
CK23        s16b        0          capmod    (15485.645,   274.438,   262.000)
CK24        s18b        0          capmod    (9397.963,    149.875,   142.000)
CK25        s7b         0          capmod    (11142.925,   214.750,   205.000)
CK26        s6b         0          capmod    (12819.750,   215.000,   211.000)
*
*
CS27        s15b        0          capmod    (3087.644,    87.750,    61.500)
CS28        cky         0          capmod    (25901.023,   646.875,   431.500)
CS29        s18b        0          capmod    (8202.481,    286.875,   191.500)
CS30        s8b         0          capmod    (17385.225,   615.000,   410.000)
CS31        s1b         0          capmod    (1495.375,    59.250,    42.500)
CS32        s2b         0          capmod    (1461.925,    59.250,    42.500)
CS33        dxn         0          capmod    (18014.338,   609.750,   411.000)
CU34        s7b         0          capmod    (12008.963,   405.750,   268.000)
CU35        s5b         0          capmod    (4213.375,    69.750,    49.500)
CW36        qx          0          capmod    (21743.162,   597.000,   401.000)
CW37        ckyn        0          capmod    (24097.201,   627.000,   422.000)
CW38        s16b        0          capmod    (6993.000,    153.750,   105.500)
CW39        qxn         0          capmod    (19240.500,   597.000,   401.000)
CW40        dx          0          capmod    (17530.025,   588.750,   397.000)
*
 .FINIS
*
 .SUBCKT fldchd (vbbx,vbby,vcs,dx,dxn,clx,clxn,cky,ckyn
+                :qx,qy,qxn,qyn)
*
 .GLOBAL vee,gnd
*
*       SUBCIRCUIT CALLS
*
   X1        (s15b,gnd)   p2k4s;* b/scdiff#1/res2d#1/r22
   X2        (s15b,gnd)   p2k4s;* b/scdiff#1/res2d#1/r21
   X3        (s24,gnd)   p2k4s;* b/scdiff#1/res2d#1/r20
   X4        (s24,gnd)   p2k4s;* b/scdiff#1/res2d#1/r19
   X5        (s16b,gnd)   p2k4s;* b/scdiff#1/res2d#1/r18
   X6        (s16b,gnd)   p2k4s;* b/scdiff#1/res2d#1/r17
   X11       (vee,s19b)   p1p51k4;* b/scdiff#1/resla#1/r8
   X13       (vee,s19b)   p1p51k4;* b/scdiff#1/resla#1/r7
   X16       (vee,s14b)   p1p51k4;* b/scdiff#1/resla#1/r16
   X17       (vee,s20b)   p1p51k4;* b/scdiff#1/resla#1/r14
   X18       (vee,s14b)   p1p51k4;* b/scdiff#1/resla#1/r15
```

```
X19       (vee,s20b)  p1p51k4;* b/scdiff#1/res1a#1/r13
X20       (vee,s26)  p1p51k4;* b/scdiff#1/res1a#1/r12
X21       (vee,s26)  p1p51k4;* b/scdiff#1/res1a#1/r11
X23       (s17b,vcs,s19b)  e1xr;* b/scdiff#1/q7
X24       (s13b,vcs,s14b)  e1xr;* b/scdiff#1/q6
X25       (s18b,vcs,s20b)  e1xr;* b/scdiff#1/q5
X26       (s25,vcs,s26)  e1xr;* b/scdiff#1/q4
X30       (s12b,ckyn,s13b)  e1xr;* b/scdiff#1/q11
X31       (qx,qx,qy)  e1xr;* b/scdiff#1/q12
X32       (gnd,s15b,s17b)  e1xr;* b/scdiff#1/q19
X33       (gnd,s15b,qx)  e1xr;* b/scdiff#1/q21
X34       (s16b,s17b,s12b)  e1xr;* b/scdiff#1/q17
X35       (s15b,s18b,s12b)  e1xr;* b/scdiff#1/q16
X36       (gnd,s16b,s18b)  e1xr;* b/scdiff#1/q18
X37       (gnd,s16b,qxn)  e1xr;* b/scdiff#1/q20
X38       (s15b,c1x,s25)  e1xr;* b/scdiff#1/q13
X39       (s15b,s7b,s11b)  e1xr;* b/scdiff#1/q14
X40       (s16b,s8b,s11b)  e1xr;* b/scdiff#1/q15
X41       (s11b,cky,s13b)  e1xr;* b/scdiff#1/q10
X42       (qxn,qxn,qyn)  e1xr;* b/scdiff#1/q8
X44       (s6b,gnd)  p2k4s;* a/scdiff#1/res2d#1/r22
X45       (s6b,gnd)  p2k4s;* a/scdiff#1/res2d#1/r21
X46       (s21,gnd)  p2k4s;* a/scdiff#1/res2d#1/r20
X47       (s21,gnd)  p2k4s;* a/scdiff#1/res2d#1/r19
X48       (s5b,gnd)  p2k4s;* a/scdiff#1/res2d#1/r18
X49       (s5b,gnd)  p2k4s;* a/scdiff#1/res2d#1/r17
X54       (vee,s10b)  p1p51k4;* a/scdiff#1/res1a#1/r8
X56       (vee,s10b)  p1p51k4;* a/scdiff#1/res1a#1/r7
X59       (vee,s9b)  p1p51k4;* a/scdiff#1/res1a#1/r16
X60       (vee,s4b)  p1p51k4;* a/scdiff#1/res1a#1/r14
X61       (vee,s9b)  p1p51k4;* a/scdiff#1/res1a#1/r15
X62       (vee,s4b)  p1p51k4;* a/scdiff#1/res1a#1/r13
X63       (vee,s23)  p1p51k4;* a/scdiff#1/res1a#1/r12
X64       (vee,s23)  p1p51k4;* a/scdiff#1/res1a#1/r11
X66       (s8b,vcs,s10b)  e1xr;* a/scdiff#1/q7
X67       (s7b,vcs,s9b)  e1xr;* a/scdiff#1/q6
X68       (s3b,vcs,s4b)  e1xr;* a/scdiff#1/q5
X69       (s22,vcs,s23)  e1xr;* a/scdiff#1/q4
X73       (s1b,cky,s3b)  e1xr;* a/scdiff#1/q11
X75       (gnd,s24,s18b)  e1xr;* a/scdiff#1/q19
X76       (gnd,s6b,s8b)  e1xr;* a/scdiff#1/q21
X77       (s24,c1xn,s25)  e1xr;* a/scdiff#1/q17
X78       (s6b,s7b,s1b)  e1xr;* a/scdiff#1/q16
X79       (gnd,s21,s7b)  e1xr;* a/scdiff#1/q18
X80       (gnd,s5b,s7b)  e1xr;* a/scdiff#1/q20
X81       (s5b,dx,s2b)  e1xr;* a/scdiff#1/q13
X82       (s6b,dxn,s2b)  e1xr;* a/scdiff#1/q14
X83       (s5b,s8b,s1b)  e1xr;* a/scdiff#1/q15
X84       (s2b,ckyn,s3b)  e1xr;* a/scdiff#1/q10
X85       (s6b,c1x,s22)  e1xr;* a/scdiff#1/q8
X86       (s21,c1xn,s22)  e1xr;* a/scdiff#1/q9
*
*       CAP/DIODE CARDS
*
*
CG1       s19b      0       capmod    (8441.550,    83.500,    79.500)
CG2       s14b      0       capmod    (4211.375,    35.000,    31.000)
CG3       s20b      0       capmod    (4676.425,    41.000,    37.000)
CG4       s10b      0       capmod    (7972.550,    83.500,    79.500)
CG5       s9b       0       capmod    (3945.000,    35.000,    31.000)
CG6       s4b       0       capmod    (4676.425,    41.000,    37.000)
CI7       qy        0       capmod    (2558.062,    39.375,    36.500)
CI8       s17b      0       capmod    (14976.912,   229.500,   233.500)
CI9       qx        0       capmod    (5042.219,    76.875,    57.000)
```

```
   CI10        s15b      0          capmod    (17441.125,   293.125,   252.000)
   CI11        s16b      0          capmod    (15645.645,   274.438,   262.000)
   CI12        s12b      0          capmod    (2752.800,    51.000,    55.000)
   CI13        ckyn      0          capmod    (14575.175,   292.250,   296.500)
   CI14        cky       0          capmod    (10720.925,   194.750,   197.000)
   CI15        s18b      0          capmod    (11635.787,   183.875,   178.000)
   CI16        s13b      0          capmod    (2677.300,    53.500,    57.500)
   CI17        s11b      0          capmod    (2431.675,    53.500,    57.500)
   CI18        s8b       0          capmod    (14112.461,   222.375,   218.500)
   CI19        s25       0          capmod    (5550.138,    92.125,    83.000)
   CI20        s7b       0          capmod    (13176.799,   236.000,   228.000)
   CI21        qxn       0          capmod    (5038.625,    76.875,    57.000)
   CI22        clx       0          capmod    (13178.299,   236.000,   239.500)
   CI23        qyn       0          capmod    (2558.062,    39.375,    36.500)
   CI24        clxn      0          capmod    (3032.925,    47.250,    41.000)
   CI25        s6b       0          capmod    (16879.975,   290.125,   278.500)
   CI26        s1b       0          capmod    (4780.538,    85.125,    76.000)
   CI27        s3b       0          capmod    (2336.050,    53.500,    57.500)
   CI28        s21       0          capmod    (7025.487,    109.750,   95.000)
   CI29        s2b       0          capmod    (4333.100,    77.000,    80.500)
   CI30        s22       0          capmod    (2979.050,    63.500,    68.500)
   CI31        dxn       0          capmod    (1635.375,    26.250,    28.000)
   CI32        dx        0          capmod    (1696.887,    27.125,    22.500)
   CK33        s24       0          capmod    (11349.499,   181.875,   172.500)
   CK34        s26       0          capmod    (3945.000,    35.000,    31.000)
   CK35        s5b       0          capmod    (12913.574,   212.125,   191.500)
   CK36        s23       0          capmod    (3945.000,    35.000,    31.000)
*
*
   CS37        s15b      0          capmod    (3087.644,    87.750,    61.500)
   CS38        cky       0          capmod    (25901.023,   646.875,   431.500)
   CS39        s18b      0          capmod    (8202.481,    286.875,   191.500)
   CS40        s8b       0          capmod    (17741.025,   615.000,   410.000)
   CS41        s25       0          capmod    (4303.938,    183.750,   125.500)
   CS42        qxn       0          capmod    (21088.775,   598.500,   402.000)
   CS43        clxn      0          capmod    (26730.350,   870.750,   579.000)
   CS44        s1b       0          capmod    (1629.175,    59.250,    42.500)
   CS45        s2b       0          capmod    (1595.725,    59.250,    42.500)
   CS46        dxn       0          capmod    (19110.762,   613.500,   408.000)
   CU47        qx        0          capmod    (20967.074,   598.500,   402.000)
   CU48        s24       0          capmod    (4338.250,    84.000,    59.000)
   CU49        s7b       0          capmod    (15361.763,   469.500,   309.000)
   CU50        s21       0          capmod    (8813.888,    239.250,   161.000)
   CU51        s5b       0          capmod    (4213.375,    69.750,    49.500)
   CW52        qy        0          capmod    (21414.588,   597.000,   401.000)
   CW53        s16b      0          capmod    (7353.750,    153.750,   105.500)
   CW54        ckyn      0          capmod    (24097.201,   627.000,   422.000)
   CW55        clx       0          capmod    (19331.199,   595.500,   400.000)
   CW56        qyn       0          capmod    (19845.525,   597.000,   401.000)
   CW57        s6b       0          capmod    (3897.925,    181.500,   124.000)
   CW58        dx        0          capmod    (18316.100,   588.750,   397.000)
*
 .FINIS
*
 .SUBCKT fldchd2 (vbbx,vbby,vcs,dx,dxn,clx1,clx1n,clx2,
+         clx2n,cky,ckyn:qx,qy,qxn,qyn)
*
 .GLOBAL vee,gnd
*
*        SUBCIRCUIT CALLS
*
   X1          (s21,gnd)   p2k4s;* flxaxd2#1/b/scdiff#1/res2d#1/r22
   X2          (s21,gnd)   p2k4s;* flxaxd2#1/b/scdiff#1/res2d#1/r21
   X3          (s21,gnd)   p2k4s;* flxaxd2#1/b/scdiff#1/res2d#1/r20
```

```
X4      (s22,gnd)   p2k4s;* f1xaxd2#1/b/scdiff#1/res2d#1/r19
X5      (s22,gnd)   p2k4s;* f1xaxd2#1/b/scdiff#1/res2d#1/r18
X6      (s22,gnd)   p2k4s;* f1xaxd2#1/b/scdiff#1/res2d#1/r17
X11     (vee,s25)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r8
X12     (vee,s26)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r6
X13     (vee,s25)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r7
X14     (vee,s26)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r5
X15     (vee,s14)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r4
X17     (vee,s19)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r14
X18     (vee,s19)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r15
X19     (vee,s19)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r13
X20     (vee,s13)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r12
X21     (vee,s13)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r11
X22     (vee,s14)   p1p51k4;* f1xaxd2#1/b/scdiff#1/res1a#1/r3
X23     (s23,vcs,s25)   e1xr;* f1xaxd2#1/b/scdiff#1/q7
X24     (s24,vcs,s26)   e1xr;* f1xaxd2#1/b/scdiff#1/q6
X25     (s17,vcs,s19)   e1xr;* f1xaxd2#1/b/scdiff#1/q5
X26     (s12,vcs,s13)   e1xr;* f1xaxd2#1/b/scdiff#1/q4
X27     (s11,vcs,s14)   e1xr;* f1xaxd2#1/b/scdiff#1/q3
X30     (s16,ckyn,s17)   e1xr;* f1xaxd2#1/b/scdiff#1/q11
X31     (qx,qx,qy)   e1xr;* f1xaxd2#1/b/scdiff#1/q12
X32     (gnd,s21,s23)   e1xr;* f1xaxd2#1/b/scdiff#1/q19
X33     (gnd,s21,qx)   e1xr;* f1xaxd2#1/b/scdiff#1/q21
X34     (s22,s23,s16)   e1xr;* f1xaxd2#1/b/scdiff#1/q17
X35     (s21,s24,s16)   e1xr;* f1xaxd2#1/b/scdiff#1/q16
X36     (gnd,s22,s24)   e1xr;* f1xaxd2#1/b/scdiff#1/q18
X37     (gnd,s22,qxn)   e1xr;* f1xaxd2#1/b/scdiff#1/q20
X39     (s21,s11,s15)   e1xr;* f1xaxd2#1/b/scdiff#1/q14
X40     (s22,s12,s15)   e1xr;* f1xaxd2#1/b/scdiff#1/q15
X41     (s15,cky,s17)   e1xr;* f1xaxd2#1/b/scdiff#1/q10
X42     (qxn,qxn,qyn)   e1xr;* f1xaxd2#1/b/scdiff#1/q8
X44     (s10,gnd)   p2k4s;* f1xaxd2#1/a/scdiff#1/res2d#1/r22
X45     (s10,gnd)   p2k4s;* f1xaxd2#1/a/scdiff#1/res2d#1/r21
X46     (s10,gnd)   p2k4s;* f1xaxd2#1/a/scdiff#1/res2d#1/r20
X47     (s9,gnd)   p2k4s;* f1xaxd2#1/a/scdiff#1/res2d#1/r19
X48     (s9,gnd)   p2k4s;* f1xaxd2#1/a/scdiff#1/res2d#1/r18
X49     (s9,gnd)   p2k4s;* f1xaxd2#1/a/scdiff#1/res2d#1/r17
X59     (vee,s8)   p1p51k4;* f1xaxd2#1/a/scdiff#1/res1a#1/r16
X60     (vee,s8)   p1p51k4;* f1xaxd2#1/a/scdiff#1/res1a#1/r14
X61     (vee,s8)   p1p51k4;* f1xaxd2#1/a/scdiff#1/res1a#1/r15
X66     (s6,vcs,s8)   e1xr;* f1xaxd2#1/a/scdiff#1/q7
X67     (s3,ckyn,s6)   e1xr;* f1xaxd2#1/a/scdiff#1/q6
X68     (s1,cky,s6)   e1xr;* f1xaxd2#1/a/scdiff#1/q5
X75     (s9,s12,s1)   e1xr;* f1xaxd2#1/a/scdiff#1/q19
X76     (gnd,s10,s12)   e1xr;* f1xaxd2#1/a/scdiff#1/q21
X79     (s10,s11,s1)   e1xr;* f1xaxd2#1/a/scdiff#1/q18
X80     (gnd,s9,s11)   e1xr;* f1xaxd2#1/a/scdiff#1/q20
X82     (s9,dx,s3)   e1xr;* f1xaxd2#1/a/scdiff#1/q14
X83     (s10,dxn,s3)   e1xr;* f1xaxd2#1/a/scdiff#1/q15
X87     (s27,gnd)   p2k4s;* c1rst2dd#1/c/scdiff#1/res2d#1/r22
X88     (s27,gnd)   p2k4s;* c1rst2dd#1/c/scdiff#1/res2d#1/r21
X102    (vee,s29)   p1p51k4;* c1rst2dd#1/c/scdiff#1/res1a#1/r16
X104    (vee,s29)   p1p51k4;* c1rst2dd#1/c/scdiff#1/res1a#1/r15
X110    (s28,vcs,s29)   e1xr;* c1rst2dd#1/c/scdiff#1/q6
X119    (gnd,s27,s11)   e1xr;* c1rst2dd#1/c/scdiff#1/q21
X120    (s10,c1x1,s28)   e1xr;* c1rst2dd#1/c/scdiff#1/q17
X121    (s27,c1x1n,s28)   e1xr;* c1rst2dd#1/c/scdiff#1/q16
X134    (s30,gnd)   p2k4s;* c1rst2dd#1/d/scdiff#1/res2d#1/r18
X135    (s30,gnd)   p2k4s;* c1rst2dd#1/d/scdiff#1/res2d#1/r17
X149    (vee,s32)   p1p51k4;* c1rst2dd#1/d/scdiff#1/res1a#1/r12
X150    (vee,s32)   p1p51k4;* c1rst2dd#1/d/scdiff#1/res1a#1/r11
X155    (s31,vcs,s32)   e1xr;* c1rst2dd#1/d/scdiff#1/q4
X165    (gnd,s30,s24)   e1xr;* c1rst2dd#1/d/scdiff#1/q18
X167    (s21,c1x2,s31)   e1xr;* c1rst2dd#1/d/scdiff#1/q13
```

```
    X168      (s30,clx2n,s31)   elxr;* clrst2dd#1/d/scdiff#1/q14
*
*       CAP/DIODE CARDS
*
*
    CE1         s25         0         capmod    (8090.050,    83.500,    79.500)
    CG2         s26         0         capmod    (9158.238,    80.375,    69.500)
    CG3         s19         0         capmod    (8238.750,    65.000,    47.000)
    CG4         s8          0         capmod    (9481.300,    83.500,    65.500)
    CG5         s29         0         capmod    (3945.000,    35.000,    31.000)
    CI6         clx2n       0         capmod    (1619.588,    23.625,    20.500)
    CI7         s31         0         capmod    (6289.650,   115.500,   119.500)
    CI8         clx2        0         capmod    (1372.088,    23.625,    20.500)
    CI9         qy          0         capmod    (2558.062,    39.375,    36.500)
    CI10        s24         0         capmod    (16917.301,   303.500,   295.500)
    CI11        s23         0         capmod    (10901.438,   198.125,   195.500)
    CI12        s16         0         capmod    (2280.300,    51.000,    55.000)
    CI13        s22         0         capmod    (16109.467,   258.750,   244.000)
    CI14        s17         0         capmod    (2255.450,    51.500,    55.500)
    CI15        s15         0         capmod    (2235.300,    51.000,    55.000)
    CI16        s11         0         capmod    (23602.641,   369.625,   367.500)
    CI17        s14         0         capmod    (8053.050,    83.500,    79.500)
    CI18        qxn         0         capmod    (5038.625,    76.875,    57.000)
    CI19        qyn         0         capmod    (2558.062,    39.375,    36.500)
    CI20        s9          0         capmod    (10502.437,   161.250,   141.500)
    CI21        s3          0         capmod    (5343.163,    93.875,    80.500)
    CI22        dxn         0         capmod    (2180.762,    35.875,    27.500)
    CI23        dx          0         capmod    (2386.287,    37.625,    28.500)
    CI24        clx1        0         capmod    (1477.875,    26.250,    22.000)
    CI25        s28         0         capmod    (6289.650,   115.500,   119.500)
    CI26        clx1n       0         capmod    (1619.588,    23.625,    20.500)
    CK27        s30         0         capmod    (6390.225,   104.500,    96.500)
    CK28        s32         0         capmod    (3945.000,    35.000,    31.000)
    CK29        s21         0         capmod    (20681.336,   340.188,   305.000)
    CK30        qx          0         capmod    (5038.625,    76.875,    57.000)
    CK31        ckyn        0         capmod    (11178.024,   211.750,   213.500)
    CK32        s12         0         capmod    (11136.075,   176.500,   166.000)
    CK33        cky         0         capmod    (11254.275,   214.250,   216.000)
    CK34        s13         0         capmod    (3945.000,    35.000,    31.000)
    CK35        s1          0         capmod    (10891.362,   174.125,   170.500)
    CK36        s6          0         capmod    (2930.700,    66.500,    70.500)
    CK37        s10         0         capmod    (28878.387,   462.750,   431.000)
    CK38        s27         0         capmod    (8590.199,   141.500,   135.500)
*
*
    CS39        clx2        0         capmod    (19728.426,   588.750,   397.000)
    CS40        s21         0         capmod    (6200.087,   129.750,    89.500)
    CS41        s24         0         capmod    (3524.688,   176.250,   120.500)
    CS42        s23         0         capmod    (1933.675,    72.000,    51.000)
    CS43        s3          0         capmod    (4080.062,   189.000,   129.000)
    CS44        clx1n       0         capmod    (22236.574,   615.000,   414.500)
    CU45        qx          0         capmod    (20484.949,   598.500,   402.000)
    CU46        ckyn        0         capmod    (19449.387,   594.750,   401.000)
    CU47        s26         0         capmod    (7643.650,   132.750,    91.500)
    CU48        s12         0         capmod    (10337.669,   339.000,   227.500)
    CU49        s11         0         capmod    (15316.338,   464.250,   314.000)
    CU50        qxn         0         capmod    (21019.488,   598.500,   402.000)
    CU51        qyn         0         capmod    (21669.824,   597.000,   401.000)
    CU52        s9          0         capmod    (15766.250,   328.500,   219.000)
    CU53        s10         0         capmod    (5499.125,    88.500,    62.000)
    CU54        clx1        0         capmod    (20716.275,   588.750,   397.000)
    CW55        clx2n       0         capmod    (21741.762,   615.000,   414.500)
    CW56        qy          0         capmod    (22584.676,   597.000,   401.000)
    CW57        s22         0         capmod    (10452.500,   216.750,   146.000)
```

```
   CW58        cky       0         capmod    (19996.850,  588.750,  397.000)
   CW59        s1        0         capmod    (6322.388,  231.750,  157.500)
   CW60        dxn       0         capmod    (23017.074,  622.500,  419.500)
   CW61        dx        0         capmod    (20073.775,  588.750,  397.000)
*
 .FINIS
*
 .SUBCKT f2dahd (vbbx,vbby,vcs,d0x,d0xn,d1x,d1xn,s0y,
+        s0yn,cky,ckyn:qx,qxn)
*
 .GLOBAL vee,gnd
*
*       SUBCIRCUIT CALLS
*
   X1        (s21,gnd)  p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r22
   X2        (s21,gnd)  p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r21
   X3        (s21,gnd)  p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r20
   X4        (s22,gnd)  p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r19
   X5        (s22,gnd)  p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r18
   X6        (s22,gnd)  p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r17
   X11       (vee,s25)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r8
   X12       (vee,s26)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r6
   X13       (vee,s25)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r7
   X14       (vee,s26)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r5
   X15       (vee,s14)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r4
   X17       (vee,s19)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r14
   X18       (vee,s19)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r15
   X19       (vee,s19)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r13
   X20       (vee,s13)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r12
   X21       (vee,s13)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r11
   X22       (vee,s14)  p1p51k4;* f2xaxd#1/b/scdiff#1/resla#1/r3
   X23       (s23,vcs,s25)  elxr;* f2xaxd#1/b/scdiff#1/q7
   X24       (s24,vcs,s26)  elxr;* f2xaxd#1/b/scdiff#1/q6
   X25       (s17,vcs,s19)  elxr;* f2xaxd#1/b/scdiff#1/q5
   X26       (s12,vcs,s13)  elxr;* f2xaxd#1/b/scdiff#1/q4
   X27       (s11,vcs,s14)  elxr;* f2xaxd#1/b/scdiff#1/q3
   X30       (s16,s18,s17)  elxr;* f2xaxd#1/b/scdiff#1/q11
   X32       (gnd,s21,s23)  elxr;* f2xaxd#1/b/scdiff#1/q19
   X33       (gnd,s21,qx)  elxr;* f2xaxd#1/b/scdiff#1/q21
   X34       (s22,s23,s16)  elxr;* f2xaxd#1/b/scdiff#1/q17
   X35       (s21,s24,s16)  elxr;* f2xaxd#1/b/scdiff#1/q16
   X36       (gnd,s22,s24)  elxr;* f2xaxd#1/b/scdiff#1/q18
   X37       (gnd,s22,qxn)  elxr;* f2xaxd#1/b/scdiff#1/q20
   X39       (s21,s11,s15)  elxr;* f2xaxd#1/b/scdiff#1/q14
   X40       (s22,s12,s15)  elxr;* f2xaxd#1/b/scdiff#1/q15
   X41       (s15,s5,s17)  elxr;* f2xaxd#1/b/scdiff#1/q10
   X43       (gnd,ckyn,s18)  elxr;* f2xaxd#1/b/scdiff#1/q9
   X44       (s10,gnd)  p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r22
   X45       (s10,gnd)  p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r21
   X46       (s10,gnd)  p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r20
   X47       (s9,gnd)  p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r19
   X48       (s9,gnd)  p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r18
   X49       (s9,gnd)  p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r17
   X54       (vee,s20)  p1p51k4;* f2xaxd#1/a/scdiff#1/resla#1/r8
   X56       (vee,s20)  p1p51k4;* f2xaxd#1/a/scdiff#1/resla#1/r7
   X58       (vee,s7)  p1p51k4;* f2xaxd#1/a/scdiff#1/resla#1/r4
   X59       (vee,s8)  p1p51k4;* f2xaxd#1/a/scdiff#1/resla#1/r16
   X60       (vee,s8)  p1p51k4;* f2xaxd#1/a/scdiff#1/resla#1/r14
   X61       (vee,s8)  p1p51k4;* f2xaxd#1/a/scdiff#1/resla#1/r15
   X65       (vee,s7)  p1p51k4;* f2xaxd#1/a/scdiff#1/resla#1/r3
   X66       (s6,vcs,s8)  elxr;* f2xaxd#1/a/scdiff#1/q7
   X67       (s4,s18,s6)  elxr;* f2xaxd#1/a/scdiff#1/q6
   X68       (s1,s5,s6)  elxr;* f2xaxd#1/a/scdiff#1/q5
   X71       (s18,vcs,s20)  elxr;* f2xaxd#1/a/scdiff#1/q2
```

```
X72       (s5,vcs,s7)  e1xr;* f2xaxd#1/a/scdiff#1/q1
X74       (gnd,cky,s5)  e1xr;* f2xaxd#1/a/scdiff#1/q12
X75       (s9,s12,s1)  e1xr;* f2xaxd#1/a/scdiff#1/q19
X76       (gnd,s10,s12)  e1xr;* f2xaxd#1/a/scdiff#1/q21
X78       (s10,d1xn,s2)  e1xr;* f2xaxd#1/a/scdiff#1/q16
X79       (s10,s11,s1)  e1xr;* f2xaxd#1/a/scdiff#1/q18
X80       (gnd,s9,s11)  e1xr;* f2xaxd#1/a/scdiff#1/q20
X81       (s9,d1x,s2)  e1xr;* f2xaxd#1/a/scdiff#1/q13
X82       (s9,d0x,s3)  e1xr;* f2xaxd#1/a/scdiff#1/q14
X83       (s10,d0xn,s3)  e1xr;* f2xaxd#1/a/scdiff#1/q15
X84       (s3,s0yn,s4)  e1xr;* f2xaxd#1/a/scdiff#1/q10
X86       (s2,s0y,s4)  e1xr;* f2xaxd#1/a/scdiff#1/q9
*
*       CAP/DIODE CARDS
*
*
CE1         s25       0         capmod    (8061.300,    83.500,    79.500)
CG2         s26       0         capmod    (9158.238,    80.375,    69.500)
CG3         s19       0         capmod    (8238.750,    65.000,    47.000)
CG4         s20       0         capmod    (6157.250,    62.500,    58.500)
CG5         s8        0         capmod    (9578.301,    83.500,    65.500)
CI6         s23       0         capmod    (10641.938,   198.125,   195.500)
CI7         s16       0         capmod    (2235.300,    51.000,    55.000)
CI8         s17       0         capmod    (2255.450,    51.500,    55.500)
CI9         s15       0         capmod    (2235.300,    51.000,    55.000)
CI10        s5        0         capmod    (17919.574,   291.500,   297.000)
CI11        s11       0         capmod    (17814.137,   274.625,   268.500)
CI12        ckyn      0         capmod    (5640.275,    109.250,   111.000)
CI13        qxn       0         capmod    (2733.450,    41.500,    45.000)
CI15        s9        0         capmod    (13423.360,   212.875,   171.000)
CI16        s1        0         capmod    (11052.362,   174.125,   170.500)
CI17        s6        0         capmod    (3110.700,    66.500,    70.500)
CI18        s10       0         capmod    (22597.361,   365.375,   326.500)
CI19        s4        0         capmod    (4591.587,    88.625,    77.500)
CI20        d1xn      0         capmod    (8137.075,    165.250,   167.000)
CI21        s2        0         capmod    (5659.650,    115.500,   119.500)
CI22        s3        0         capmod    (4556.325,    87.750,    77.000)
CI23        d0xn      0         capmod    (2180.762,    35.875,    27.500)
CI24        s0yn      0         capmod    (5317.875,    101.250,   103.000)
CI25        s7        0         capmod    (6437.875,    62.500,    58.500)
CI26        d0x       0         capmod    (2386.287,    37.625,    28.500)
CI27        s0y       0         capmod    (3065.675,    52.250,    54.000)
CI28        d1x       0         capmod    (2022.475,    33.250,    35.000)
CK29        qx        0         capmod    (2733.450,    41.500,    45.000)
CK30        s22       0         capmod    (16063.467,   258.750,   244.000)
CK31        s21       0         capmod    (14302.650,   237.688,   202.500)
CK32        s18       0         capmod    (13674.787,   252.938,   248.000)
CK33        s24       0         capmod    (10405.150,   175.500,   167.000)
CK34        s12       0         capmod    (11067.075,   176.500,   166.000)
CK35        s13       0         capmod    (3945.000,    35.000,    31.000)
CK36        s14       0         capmod    (8024.300,    83.500,    79.500)
*
*
CQ37        s18       0         capmod    (6076.600,    119.250,   82.500)
CS38        s23       0         capmod    (1799.875,    72.000,    51.000)
CS39        s21       0         capmod    (6200.087,    129.750,   89.500)
CS40        s24       0         capmod    (3524.688,    176.250,   120.500)
CS41        s1        0         capmod    (6373.038,    231.750,   157.500)
CS42        s4        0         capmod    (2389.900,    111.000,   77.000)
CS43        s3        0         capmod    (1629.175,    59.250,    42.500)
CS44        s0y       0         capmod    (20442.100,   588.750,   397.000)
CU45        s26       0         capmod    (7643.650,    132.750,   91.500)
CU46        s12       0         capmod    (10337.669,   339.000,   227.500)
CU47        s5        0         capmod    (5240.025,    98.250,    68.500)
```

```
   CU48        s11       0        capmod    (15773.487,   464.250,   314.000)
   CU49        ckyn      0        capmod    (20404.699,   594.750,   401.000)
   CU50        qxn       0        capmod    (19754.949,   597.000,   401.000)
   CU51        s9        0        capmod    (15544.250,   328.500,   219.000)
   CU52        s10       0        capmod    (5637.875,    91.500,    64.000)
   CU53        d0xn      0        capmod    (24051.963,   622.500,   419.500)
   CW54        qx        0        capmod    (20180.199,   597.000,   401.000)
   CW55        s22       0        capmod    (10452.500,   216.750,   146.000)
   CW57        d1xn      0        capmod    (17513.301,   588.750,   397.000)
   CW58        s0yn      0        capmod    (19325.650,   588.750,   397.000)
   CW59        d0x       0        capmod    (22126.574,   588.750,   397.000)
   CW60        d1x       0        capmod    (17647.100,   588.750,   397.000)
*
 .FINIS
*
 .SUBCKT f2dchd (vbbx,vbby,vcs,d0x,d0xn,d1x,d1xn,s0y,s0yn,c1x1,
+          c1x1n,c1x2,c1x2n,cky,
+          ckyn:qx,qy,qxn,qyn)
*
 .GLOBAL vee,gnd
*
*        SUBCIRCUIT CALLS
*
   X1          (s21,gnd)   p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r22
   X2          (s21,gnd)   p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r21
   X3          (s21,gnd)   p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r20
   X4          (s22,gnd)   p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r19
   X5          (s22,gnd)   p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r18
   X6          (s22,gnd)   p2k4s;* f2xaxd#1/b/scdiff#1/res2d#1/r17
   X11         (vee,s25)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r8
   X12         (vee,s26)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r6
   X13         (vee,s25)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r7
   X14         (vee,s26)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r5
   X15         (vee,s14)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r4
   X17         (vee,s19)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r14
   X18         (vee,s19)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r15
   X19         (vee,s19)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r13
   X20         (vee,s13)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r12
   X21         (vee,s13)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r11
   X22         (vee,s14)   p1p51k4;* f2xaxd#1/b/scdiff#1/res1a#1/r3
   X23         (s23,vcs,s25)   e1xr;* f2xaxd#1/b/scdiff#1/q7
   X24         (s24,vcs,s26)   e1xr;* f2xaxd#1/b/scdiff#1/q6
   X25         (s17,vcs,s19)   e1xr;* f2xaxd#1/b/scdiff#1/q5
   X26         (s12,vcs,s13)   e1xr;* f2xaxd#1/b/scdiff#1/q4
   X27         (s11,vcs,s14)   e1xr;* f2xaxd#1/b/scdiff#1/q3
   X30         (s16,s18,s17)   e1xr;* f2xaxd#1/b/scdiff#1/q11
   X31         (qx,qx,qy)   e1xr;* f2xaxd#1/b/scdiff#1/q12
   X32         (gnd,s21,s23)   e1xr;* f2xaxd#1/b/scdiff#1/q19
   X33         (gnd,s21,qx)   e1xr;* f2xaxd#1/b/scdiff#1/q21
   X34         (s22,s23,s16)   e1xr;* f2xaxd#1/b/scdiff#1/q17
   X35         (s21,s24,s16)   e1xr;* f2xaxd#1/b/scdiff#1/q16
   X36         (gnd,s22,s24)   e1xr;* f2xaxd#1/b/scdiff#1/q18
   X37         (gnd,s22,qxn)   e1xr;* f2xaxd#1/b/scdiff#1/q20
   X39         (s21,s11,s15)   e1xr;* f2xaxd#1/b/scdiff#1/q14
   X40         (s22,s12,s15)   e1xr;* f2xaxd#1/b/scdiff#1/q15
   X41         (s15,s5,s17)   e1xr;* f2xaxd#1/b/scdiff#1/q10
   X42         (qxn,qxn,qyn)   e1xr;* f2xaxd#1/b/scdiff#1/q8
   X43         (gnd,ckyn,s18)   e1xr;* f2xaxd#1/b/scdiff#1/q9
   X44         (s10,gnd)   p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r22
   X45         (s10,gnd)   p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r21
   X46         (s10,gnd)   p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r20
   X47         (s9,gnd)   p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r19
   X48         (s9,gnd)   p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r18
   X49         (s9,gnd)   p2k4s;* f2xaxd#1/a/scdiff#1/res2d#1/r17
```

```
X54       (vee,s20)   p1p51k4;* f2xaxd#1/a/scdiff#1/res1a#1/r8
X56       (vee,s20)   p1p51k4;* f2xaxd#1/a/scdiff#1/res1a#1/r7
X58       (vee,s7)   p1p51k4;* f2xaxd#1/a/scdiff#1/res1a#1/r4
X59       (vee,s8)   p1p51k4;* f2xaxd#1/a/scdiff#1/res1a#1/r16
X60       (vee,s8)   p1p51k4;* f2xaxd#1/a/scdiff#1/res1a#1/r14
X61       (vee,s8)   p1p51k4;* f2xaxd#1/a/scdiff#1/res1a#1/r15
X65       (vee,s7)   p1p51k4;* f2xaxd#1/a/scdiff#1/res1a#1/r3
X66       (s6,vcs,s8)   e1xr;* f2xaxd#1/a/scdiff#1/q7
X67       (s4,s18,s6)   e1xr;* f2xaxd#1/a/scdiff#1/q6
X68       (s1,s5,s6)   e1xr;* f2xaxd#1/a/scdiff#1/q5
X71       (s18,vcs,s20)   e1xr;* f2xaxd#1/a/scdiff#1/q2
X72       (s5,vcs,s7)   e1xr;* f2xaxd#1/a/scdiff#1/q1
X74       (gnd,cky,s5)   e1xr;* f2xaxd#1/a/scdiff#1/q12
X75       (s9,s12,s1)   e1xr;* f2xaxd#1/a/scdiff#1/q19
X76       (gnd,s10,s12)   e1xr;* f2xaxd#1/a/scdiff#1/q21
X78       (s10,d1xn,s2)   e1xr;* f2xaxd#1/a/scdiff#1/q16
X79       (s10,s11,s1)   e1xr;* f2xaxd#1/a/scdiff#1/q18
X80       (gnd,s9,s11)   e1xr;* f2xaxd#1/a/scdiff#1/q20
X81       (s9,d1x,s2)   e1xr;* f2xaxd#1/a/scdiff#1/q13
X82       (s9,d0x,s3)   e1xr;* f2xaxd#1/a/scdiff#1/q14
X83       (s10,d0xn,s3)   e1xr;* f2xaxd#1/a/scdiff#1/q15
X84       (s3,s0yn,s4)   e1xr;* f2xaxd#1/a/scdiff#1/q10
X86       (s2,s0y,s4)   e1xr;* f2xaxd#1/a/scdiff#1/q9
X87       (s27,gnd)   p2k4s;* clrst2dd#1/c/scdiff#1/res2d#1/r22
X88       (s27,gnd)   p2k4s;* clrst2dd#1/c/scdiff#1/res2d#1/r21
X102      (vee,s29)   p1p51k4;* clrst2dd#1/c/scdiff#1/res1a#1/r16
X104      (vee,s29)   p1p51k4;* clrst2dd#1/c/scdiff#1/res1a#1/r15
X110      (s28,vcs,s29)   e1xr;* clrst2dd#1/c/scdiff#1/q6
X119      (gnd,s27,s11)   e1xr;* clrst2dd#1/c/scdiff#1/q21
X120      (s10,c1x1,s28)   e1xr;* clrst2dd#1/c/scdiff#1/q17
X121      (s27,c1x1n,s28)   e1xr;* clrst2dd#1/c/scdiff#1/q16
X134      (s30,gnd)   p2k4s;* clrst2dd#1/d/scdiff#1/res2d#1/r18
X135      (s30,gnd)   p2k4s;* clrst2dd#1/d/scdiff#1/res2d#1/r17
X149      (vee,s32)   p1p51k4;* clrst2dd#1/d/scdiff#1/res1a#1/r12
X150      (vee,s32)   p1p51k4;* clrst2dd#1/d/scdiff#1/res1a#1/r11
X155      (s31,vcs,s32)   e1xr;* clrst2dd#1/d/scdiff#1/q4
X165      (gnd,s30,s24)   e1xr;* clrst2dd#1/d/scdiff#1/q18
X167      (s21,c1x2,s31)   e1xr;* clrst2dd#1/d/scdiff#1/q13
X168      (s30,c1x2n,s31)   e1xr;* clrst2dd#1/d/scdiff#1/q14
*
*       CAP/DIODE CARDS
*
*
CE1          s25          0          capmod    (8090.050,    83.500,    79.500)
CG2          s26          0          capmod    (9158.238,    80.375,    69.500)
CG3          s29          0          capmod    (3945.000,    35.000,    31.000)
CG4          s8           0          capmod    (9578.301,    83.500,    65.500)
CG5          s20          0          capmod    (6157.250,    62.500,    58.500)
CG6          s19          0          capmod    (8238.750,    65.000,    47.000)
CI7          s10          0          capmod    (29726.186,   476.875,   442.000)
CI8          c1x1n        0          capmod    (1619.588,    23.625,    20.500)
CI9          c1x1         0          capmod    (1609.125,    26.250,    22.000)
CI10         d1xn         0          capmod    (8137.075,    165.250,   167.000)
CI11         d1x          0          capmod    (2022.475,    33.250,    35.000)
CI12         s0yn         0          capmod    (5317.875,    101.250,   103.000)
CI13         s0y          0          capmod    (3065.675,    52.250,    54.000)
CI14         d0x          0          capmod    (2386.287,    37.625,    28.500)
CI15         d0xn         0          capmod    (2180.762,    35.875,    27.500)
CI16         s9           0          capmod    (13423.360,   212.875,   171.000)
CI17         s5           0          capmod    (18129.574,   291.500,   297.000)
CI18         s1           0          capmod    (11052.362,   174.125,   170.500)
CI19         s4           0          capmod    (4591.587,    88.625,    77.500)
CI20         s3           0          capmod    (4556.325,    87.750,    77.000)
CI21         s11          0          capmod    (24285.641,   369.625,   367.500)
```

```
CI22        ckyn      0        capmod    (5820.275,   109.250,   111.000)
CI24        qyn       0        capmod    (2558.062,    39.375,    36.500)
CI25        qxn       0        capmod    (5038.625,    76.875,    57.000)
CI26        s22       0        capmod    (16109.467,  258.750,   244.000)
CI27        s24       0        capmod    (16917.301,  303.500,   295.500)
CI28        s23       0        capmod    (10901.438,  198.125,   195.500)
CI29        qy        0        capmod    (2558.062,    39.375,    36.500)
CI30        clx2      0        capmod    (1372.088,    23.625,    20.500)
CI31        clx2n     0        capmod    (1619.588,    23.625,    20.500)
CI32        s28       0        capmod    (6364.650,   115.500,   119.500)
CI33        s7        0        capmod    (6437.875,    62.500,    58.500)
CI34        s2        0        capmod    (5659.650,   115.500,   119.500)
CI35        s6        0        capmod    (3110.700,    66.500,    70.500)
CI36        s14       0        capmod    (8053.050,    83.500,    79.500)
CI37        s15       0        capmod    (2235.300,    51.000,    55.000)
CI38        s17       0        capmod    (2255.450,    51.500,    55.500)
CI39        s16       0        capmod    (2280.300,    51.000,    55.000)
CI40        s31       0        capmod    (6289.650,   115.500,   119.500)
CK41        s18       0        capmod    (13854.787,  252.938,   248.000)
CK42        s12       0        capmod    (11136.075,  176.500,   166.000)
CK43        s21       0        capmod    (20681.336,  340.188,   305.000)
CK44        qx        0        capmod    (5038.625,    76.875,    57.000)
CK45        s27       0        capmod    (8590.199,   141.500,   135.500)
CK46        s13       0        capmod    (3945.000,    35.000,    31.000)
CK47        s30       0        capmod    (6390.225,   104.500,    96.500)
CK48        s32       0        capmod    (3945.000,    35.000,    31.000)
*
*
CQ49        s18       0        capmod    (6076.600,   119.250,    82.500)
CS50        clx1n     0        capmod    (22236.574,  615.000,   414.500)
CS51        d1xn      0        capmod    (17951.082,  588.750,   397.000)
CS52        s4        0        capmod    (2389.900,   111.000,    77.000)
CS53        s3        0        capmod    (1629.175,    59.250,    42.500)
CS54        s21       0        capmod    (6200.087,   129.750,    89.500)
CS55        s24       0        capmod    (3524.688,   176.250,   120.500)
CS56        s23       0        capmod    (1933.675,    72.000,    51.000)
CS57        clx2      0        capmod    (19728.426,  588.750,   397.000)
CU58        s10       0        capmod    (5637.875,    91.500,    64.000)
CU59        clx1      0        capmod    (20716.275,  588.750,   397.000)
CU60        s0y       0        capmod    (20738.100,  588.750,   397.000)
CU61        d0xn      0        capmod    (24273.963,  622.500,   419.500)
CU62        s9        0        capmod    (15766.250,  328.500,   219.000)
CU63        s5        0        capmod    (5240.025,    98.250,    68.500)
CU64        s11       0        capmod    (15773.487,  464.250,   314.000)
CU65        ckyn      0        capmod    (20404.699,  594.750,   401.000)
CU66        qyn       0        capmod    (21803.625,  597.000,   401.000)
CU67        qxn       0        capmod    (21153.287,  598.500,   402.000)
CU68        s12       0        capmod    (10337.669,  339.000,   227.500)
CU69        s26       0        capmod    (7643.650,   132.750,    91.500)
CU70        qx        0        capmod    (20484.949,  598.500,   402.000)
CW71        d1x       0        capmod    (18091.100,  588.750,   397.000)
CW72        s0yn      0        capmod    (20065.650,  588.750,   397.000)
CW73        d0x       0        capmod    (22348.574,  588.750,   397.000)
CW74        s1        0        capmod    (6909.788,   231.750,   157.500)
CW76        s22       0        capmod    (10452.500,  216.750,   146.000)
CW77        qy        0        capmod    (22584.676,  597.000,   401.000)
CW78        clx2n     0        capmod    (21741.762,  615.000,   414.500)
*
 .FINIS
*
 .SUBCKT f3dahd (vbbx,vbby,vcs,dx,dxn,ey,eyn:qx,qxn)
*
 .GLOBAL vee,gnd
*
```

```
*       SUBCIRCUIT CALLS
*
X1      (s5a,gnd)  p2k4s;* a/scdiff#1/res2d#1/r22
X2      (s5a,gnd)  p2k4s;* a/scdiff#1/res2d#1/r21
X5      (s6a,gnd)  p2k4s;* a/scdiff#1/res2d#1/r18
X6      (s6a,gnd)  p2k4s;* a/scdiff#1/res2d#1/r17
X11     (vee,s9a)  p1p51k4;* a/scdiff#1/res1a#1/r8
X13     (vee,s9a)  p1p51k4;* a/scdiff#1/res1a#1/r7
X16     (vee,s4a)  p1p51k4;* a/scdiff#1/res1a#1/r16
X17     (vee,s10a)  p1p51k4;* a/scdiff#1/res1a#1/r14
X18     (vee,s4a)  p1p51k4;* a/scdiff#1/res1a#1/r15
X19     (vee,s10a)  p1p51k4;* a/scdiff#1/res1a#1/r13
X23     (s7a,vcs,s9a)  e1xr;* a/scdiff#1/q7
X24     (s3a,vcs,s4a)  e1xr;* a/scdiff#1/q6
X25     (s8a,vcs,s10a)  e1xr;* a/scdiff#1/q5
X30     (s1a,eyn,s3a)  e1xr;* a/scdiff#1/q11
X32     (gnd,s5a,s7a)  e1xr;* a/scdiff#1/q19
X33     (gnd,s5a,qx)  e1xr;* a/scdiff#1/q21
X35     (s6a,s7a,s1a)  e1xr;* a/scdiff#1/q16
X36     (gnd,s6a,s8a)  e1xr;* a/scdiff#1/q18
X37     (gnd,s6a,qxn)  e1xr;* a/scdiff#1/q20
X38     (s6a,dx,s2a)  e1xr;* a/scdiff#1/q13
X39     (s5a,dxn,s2a)  e1xr;* a/scdiff#1/q14
X40     (s5a,s8a,s1a)  e1xr;* a/scdiff#1/q15
X41     (s2a,ey,s3a)  e1xr;* a/scdiff#1/q10
*
*       CAP/DIODE CARDS
*
*
CG1         s9a         0         capmod    (8357.300,    83.500,    79.500)
CG2         s4a         0         capmod    (4495.475,    37.000,    33.000)
CG3         s10a        0         capmod    (4340.012,    36.500,    32.500)
CI4         qx          0         capmod    (2734.887,    41.500,    45.000)
CI5         s7a         0         capmod    (6624.950,   113.375,   106.000)
CI6         s5a         0         capmod    (13394.361,  223.500,   220.500)
CI7         s1a         0         capmod    (4780.200,    84.000,    87.500)
CI8         ey          0         capmod    (3224.975,    58.250,    60.000)
CI9         s3a         0         capmod    (2275.150,    50.500,    55.500)
CI10        eyn         0         capmod    (1503.337,    23.625,    20.500)
CI11        s6a         0         capmod    (19619.361,  346.000,   340.500)
CI12        s2a         0         capmod    (4827.888,    84.625,    81.000)
CI13        dxn         0         capmod    (1503.337,    23.625,    20.500)
CI15        qxn         0         capmod    (2734.887,    41.500,    45.000)
CK16        s8a         0         capmod    (7861.637,   124.625,   125.000)
*
*
CS17        s1a         0         capmod    (1428.850,    57.000,    41.000)
CS18        s8a         0         capmod    (6717.150,   253.125,   165.500)
CS19        s2a         0         capmod    (1434.550,    56.250,    40.500)
CU21        qx          0         capmod    (18617.100,  597.000,   401.000)
CU22        ey          0         capmod    (22624.699,  597.750,   401.000)
CU23        s6a         0         capmod    (3422.500,    60.000,    43.000)
CW24        s7a         0         capmod    (6034.100,   244.500,   166.000)
CW25        eyn         0         capmod    (24186.549,  594.375,   399.500)
CW26        dxn         0         capmod    (22768.674,  597.375,   401.500)
CW27        qxn         0         capmod    (19018.500,  597.000,   401.000)
*
.FINIS
*
.SUBCKT f4dahd (vbbx,vbby,vcs,d0x,d0xn,d1x,d1xn,
+          s0y,s0yn,ey,eyn:qx,qxn)
*
.GLOBAL vee,gnd
*
```

```
*       SUBCIRCUIT CALLS
*
X1      (s11,gnd)  p2k4s;* b/scdiff#1/res2d#1/r22
X2      (s11,gnd)  p2k4s;* b/scdiff#1/res2d#1/r21
X5      (s12,gnd)  p2k4s;* b/scdiff#1/res2d#1/r18
X6      (s12,gnd)  p2k4s;* b/scdiff#1/res2d#1/r17
X15     (vee,s8)  p1p51k4;* b/scdiff#1/res1a#1/r4
X16     (vee,s15)  p1p51k4;* b/scdiff#1/res1a#1/r16
X18     (vee,s15)  p1p51k4;* b/scdiff#1/res1a#1/r15
X20     (vee,s16)  p1p51k4;* b/scdiff#1/res1a#1/r12
X21     (vee,s16)  p1p51k4;* b/scdiff#1/res1a#1/r11
X22     (vee,s8)  p1p51k4;* b/scdiff#1/res1a#1/r3
X24     (s13,vcs,s15)  e1xr;* b/scdiff#1/q6
X26     (s14,vcs,s16)  e1xr;* b/scdiff#1/q4
X27     (s5,vcs,s8)  e1xr;* b/scdiff#1/q3
X32     (gnd,s11,s13)  e1xr;* b/scdiff#1/q19
X33     (gnd,s11,qx)  e1xr;* b/scdiff#1/q21
X36     (gnd,s12,s14)  e1xr;* b/scdiff#1/q18
X37     (gnd,s12,qxn)  e1xr;* b/scdiff#1/q20
X39     (s11,s14,s1)  e1xr;* b/scdiff#1/q14
X40     (s12,s13,s1)  e1xr;* b/scdiff#1/q15
X42     (gnd,eyn,s5)  e1xr;* b/scdiff#1/q8
X43     (s1,s5,s6)  e1xr;* b/scdiff#1/q9
X54     (vee,s9)  p1p51k4;* a/scdiff#1/res1a#1/r8
X56     (vee,s9)  p1p51k4;* a/scdiff#1/res1a#1/r7
X59     (vee,s10)  p1p51k4;* a/scdiff#1/res1a#1/r16
X61     (vee,s10)  p1p51k4;* a/scdiff#1/res1a#1/r15
X66     (s6,vcs,s9)  e1xr;* a/scdiff#1/q7
X67     (s7,vcs,s10)  e1xr;* a/scdiff#1/q6
X73     (gnd,ey,s7)  e1xr;* a/scdiff#1/q11
X74     (s4,s7,s6)  e1xr;* a/scdiff#1/q12
X77     (s11,d0xn,s3)  e1xr;* a/scdiff#1/q17
X78     (s12,d0x,s3)  e1xr;* a/scdiff#1/q16
X82     (s11,d1xn,s2)  e1xr;* a/scdiff#1/q14
X83     (s12,d1x,s2)  e1xr;* a/scdiff#1/q15
X84     (s3,s0yn,s4)  e1xr;* a/scdiff#1/q10
X86     (s2,s0y,s4)  e1xr;* a/scdiff#1/q9
*
*       CAP/DIODE CARDS
*
*
CG1       s15      0       capmod   (3945.000,   35.000,   31.000)
CG2       s9       0       capmod   (7882.550,   83.500,   79.500)
CG3       s10      0       capmod   (3945.000,   35.000,   31.000)
CI4       qx       0       capmod   (2734.887,   41.500,   45.000)
CI5       s11      0       capmod   (22224.699,  360.250,  344.000)
CI6       s13      0       capmod   (8279.050,  148.500,  137.000)
CI7       s1       0       capmod   (2215.150,   50.500,   55.500)
CI8       s6       0       capmod   (5293.275,  105.500,  109.500)
CI9       qxn      0       capmod   (2734.887,   41.500,   45.000)
CI10      eyn      0       capmod   (4758.925,   79.750,   81.500)
CI11      ey       0       capmod   (3532.425,   62.250,   64.000)
CI12      s4       0       capmod   (5292.700,  109.000,  113.000)
CI13      s7       0       capmod   (2876.950,   56.500,   62.500)
CI14      d0xn     0       capmod   (1609.125,   26.250,   22.000)
CI15      s3       0       capmod   (3794.075,   74.000,   80.000)
CI17      d1x      0       capmod   (1513.137,   27.125,   22.500)
CI18      s2       0       capmod   (2395.150,   50.500,   55.500)
CI19      s0yn     0       capmod   (3185.575,   60.250,   62.000)
CI20      d1xn     0       capmod   (1850.637,   27.125,   22.500)
CI21      s0y      0       capmod   (3297.750,   67.500,   69.250)
CK22      s12      0       capmod   (21573.023,  353.625,  340.000)
CK23      s14      0       capmod   (4582.962,   77.375,   75.500)
CK24      s5       0       capmod   (5562.300,  113.500,  117.500)
```

```
    CK25        s16         0        capmod    (3945.000,   35.000,   31.000)
    CK26        s8          0        capmod    (8024.300,   83.500,   79.500)
*
*
    CS27        s11         0        capmod    (7679.025,  159.750,  109.500)
    CS28        s13         0        capmod    (3138.350,  169.500,  116.000)
    CS30        d1xn        0        capmod    (22074.625,  618.000,  416.500)
    CU31        qx          0        capmod    (18617.100,  597.000,  401.000)
    CU32        s12         0        capmod    (2976.350,   56.250,   40.500)
    CU33        d0xn        0        capmod    (20545.412,  591.750,  399.000)
    CU34        d1x         0        capmod    (21371.426,  588.750,  397.000)
    CW35        s14         0        capmod    (7705.350,  282.750,  190.000)
    CW36        qxn         0        capmod    (19914.074,  597.000,  401.000)
    CW37        eyn         0        capmod    (19497.750,  588.750,  397.000)
    CW38        ey          0        capmod    (18224.900,  588.750,  397.000)
    CW39        s0yn        0        capmod    (18044.000,  615.750,  415.000)
    CW40        s0y         0        capmod    (17513.301,  588.750,  397.000)
*
  .FINIS
*
*
  .SUBCKT f5dahd (vbbx,vbby,vcs,dx,dxn,ey,eyn,cky,ckyn:qx,qxn)
*
  .GLOBAL vee,gnd
*
*         SUBCIRCUIT CALLS
*
    X1          (s15b,gnd)   p2k4s;* c/scdiff#1/res2d#1/r22
    X2          (s15b,gnd)   p2k4s;* c/scdiff#1/res2d#1/r21
    X5          (s16b,gnd)   p2k4s;* c/scdiff#1/res2d#1/r18
    X6          (s16b,gnd)   p2k4s;* c/scdiff#1/res2d#1/r17
    X11         (vee,s19b)   p1p51k4;* c/scdiff#1/res1a#1/r8
    X13         (vee,s19b)   p1p51k4;* c/scdiff#1/res1a#1/r7
    X16         (vee,s14b)   p1p51k4;* c/scdiff#1/res1a#1/r16
    X17         (vee,s20b)   p1p51k4;* c/scdiff#1/res1a#1/r14
    X18         (vee,s14b)   p1p51k4;* c/scdiff#1/res1a#1/r15
    X19         (vee,s20b)   p1p51k4;* c/scdiff#1/res1a#1/r13
    X23         (s17b,vcs,s19b)   e1xr;* c/scdiff#1/q7
    X24         (s13b,vcs,s14b)   e1xr;* c/scdiff#1/q6
    X25         (s18b,vcs,s20b)   e1xr;* c/scdiff#1/q5
    X30         (s12b,ckyn,s13b)   e1xr;* c/scdiff#1/q11
    X32         (gnd,s15b,s17b)   e1xr;* c/scdiff#1/q19
    X33         (gnd,s15b,qx)   e1xr;* c/scdiff#1/q21
    X34         (s16b,s17b,s12b)   e1xr;* c/scdiff#1/q17
    X35         (s15b,s18b,s12b)   e1xr;* c/scdiff#1/q16
    X36         (gnd,s16b,s18b)   e1xr;* c/scdiff#1/q18
    X37         (gnd,s16b,qxn)   e1xr;* c/scdiff#1/q20
    X39         (s15b,s7b,s11b)   e1xr;* c/scdiff#1/q14
    X40         (s16b,s8b,s11b)   e1xr;* c/scdiff#1/q15
    X41         (s11b,cky,s13b)   e1xr;* c/scdiff#1/q10
    X44         (s6b,gnd)   p2k4s;* b/scdiff#1/res2d#1/r22
    X45         (s6b,gnd)   p2k4s;* b/scdiff#1/res2d#1/r21
    X48         (s5b,gnd)   p2k4s;* b/scdiff#1/res2d#1/r18
    X49         (s5b,gnd)   p2k4s;* b/scdiff#1/res2d#1/r17
    X54         (vee,s10b)   p1p51k4;* b/scdiff#1/res1a#1/r8
    X56         (vee,s10b)   p1p51k4;* b/scdiff#1/res1a#1/r7
    X59         (vee,s9b)   p1p51k4;* b/scdiff#1/res1a#1/r16
    X60         (vee,s4b)   p1p51k4;* b/scdiff#1/res1a#1/r14
    X61         (vee,s9b)   p1p51k4;* b/scdiff#1/res1a#1/r15
    X62         (vee,s4b)   p1p51k4;* b/scdiff#1/res1a#1/r13
    X66         (s8b,vcs,s10b)   e1xr;* b/scdiff#1/q7
    X67         (s7b,vcs,s9b)   e1xr;* b/scdiff#1/q6
    X68         (s3b,vcs,s4b)   e1xr;* b/scdiff#1/q5
    X73         (s1b,cky,s3b)   e1xr;* b/scdiff#1/q11
    X76         (gnd,s6b,s8b)   e1xr;* b/scdiff#1/q21
```

```
X78       (s6b,s7b,s1b)   e1xr;* b/scdiff#1/q16
X80       (gnd,s5b,s7b)   e1xr;* b/scdiff#1/q20
X81       (s5b,s7a,s2b)   e1xr;* b/scdiff#1/q13
X82       (s6b,s8a,s2b)   e1xr;* b/scdiff#1/q14
X83       (s5b,s8b,s1b)   e1xr;* b/scdiff#1/q15
X84       (s2b,ckyn,s3b)   e1xr;* b/scdiff#1/q10
X87       (s5a,gnd)   p2k4s;* a/scdiff#1/res2d#1/r22
X88       (s5a,gnd)   p2k4s;* a/scdiff#1/res2d#1/r21
X91       (s6a,gnd)   p2k4s;* a/scdiff#1/res2d#1/r18
X92       (s6a,gnd)   p2k4s;* a/scdiff#1/res2d#1/r17
X97       (vee,s9a)   p1p51k4;* a/scdiff#1/res1a#1/r8
X99       (vee,s9a)   p1p51k4;* a/scdiff#1/res1a#1/r7
X102      (vee,s4a)   p1p51k4;* a/scdiff#1/res1a#1/r16
X103      (vee,s10a)   p1p51k4;* a/scdiff#1/res1a#1/r14
X104      (vee,s4a)   p1p51k4;* a/scdiff#1/res1a#1/r15
X105      (vee,s10a)   p1p51k4;* a/scdiff#1/res1a#1/r13
X109      (s7a,vcs,s9a)   e1xr;* a/scdiff#1/q7
X110      (s3a,vcs,s4a)   e1xr;* a/scdiff#1/q6
X111      (s8a,vcs,s10a)   e1xr;* a/scdiff#1/q5
X116      (s1a,eyn,s3a)   e1xr;* a/scdiff#1/q11
X118      (gnd,s5a,s7a)   e1xr;* a/scdiff#1/q19
X121      (s6a,s7a,s1a)   e1xr;* a/scdiff#1/q16
X122      (gnd,s6a,s8a)   e1xr;* a/scdiff#1/q18
X124      (s6a,dx,s2a)   e1xr;* a/scdiff#1/q13
X125      (s5a,dxn,s2a)   e1xr;* a/scdiff#1/q14
X126      (s5a,s8a,s1a)   e1xr;* a/scdiff#1/q15
X127      (s2a,ey,s3a)   e1xr;* a/scdiff#1/q10
*
*       CAP/DIODE CARDS
*
*
CG1       s19b      0       capmod   (8412.800,   83.500,   79.500)
CG2       s14b      0       capmod   (4211.375,   35.000,   31.000)
CG3       s20b      0       capmod   (4676.425,   41.000,   37.000)
CG4       s10b      0       capmod   (7943.800,   83.500,   79.500)
CG5       s9b       0       capmod   (3945.000,   35.000,   31.000)
CG6       s4b       0       capmod   (4676.425,   41.000,   37.000)
CG7       s9a       0       capmod   (8276.800,   83.500,   79.500)
CG8       s4a       0       capmod   (4495.475,   37.000,   33.000)
CG9       s10a      0       capmod   (4340.012,   36.500,   32.500)
CI10      qx        0       capmod   (2734.887,   41.500,   45.000)
CI11      s17b      0       capmod   (14115.662,  229.500,  233.500)
CI12      ckyn      0       capmod   (13532.676,  292.250,  296.500)
CI13      s12b      0       capmod   (2707.800,   51.000,   55.000)
CI14      cky       0       capmod   (9655.925,   194.750,  197.000)
CI15      s13b      0       capmod   (2677.300,   53.500,   57.500)
CI16      s11b      0       capmod   (2431.675,   53.500,   57.500)
CI17      s8b       0       capmod   (13796.212,  222.375,  218.500)
CI18      qxn       0       capmod   (2733.450,   41.500,   45.000)
CI19      s1b       0       capmod   (4780.538,   85.125,   76.000)
CI20      s3b       0       capmod   (2336.050,   53.500,   57.500)
CI21      s5b       0       capmod   (12759.763,  212.125,  191.500)
CI22      s2b       0       capmod   (4078.100,   77.000,   80.500)
CI23      s7a       0       capmod   (8102.825,   139.625,  128.000)
CI24      s5a       0       capmod   (9559.012,   161.500,  158.500)
CI25      s1a       0       capmod   (4780.200,   84.000,   87.500)
CI26      ey        0       capmod   (3224.975,   58.250,   60.000)
CI27      s3a       0       capmod   (2275.150,   50.500,   55.500)
CI28      eyn       0       capmod   (1503.337,   23.625,   20.500)
CI29      s2a       0       capmod   (4827.888,   84.625,   81.000)
CI30      dxn       0       capmod   (1503.337,   23.625,   20.500)
CI31      dx        0       capmod   (1372.088,   23.625,   20.500)
CK32      s15b      0       capmod   (16161.275,  263.625,  222.500)
CK33      s16b      0       capmod   (15485.645,  274.438,  262.000)
```

```
    CK34        s18b        0          capmod    (9397.963,   149.875,   142.000)
    CK35        s7b         0          capmod    (11142.925,  214.750,   205.000)
    CK36        s6b         0          capmod    (12819.750,  215.000,   211.000)
    CK37        s8a         0          capmod    (13995.561,  229.375,   231.000)
    CK38        s6a         0          capmod    (15244.012,  284.000,   278.500)
*
*
    CS39        s15b        0          capmod    (3087.644,   87.750,    61.500)
    CS40        cky         0          capmod    (25901.023,  646.875,   431.500)
    CS41        s18b        0          capmod    (8202.481,   286.875,   191.500)
    CS42        s8b         0          capmod    (17385.225,  615.000,   410.000)
    CS43        s1b         0          capmod    (1495.375,   59.250,    42.500)
    CS44        s2b         0          capmod    (1461.925,   59.250,    42.500)
    CS45        s8a         0          capmod    (9997.500,   406.875,   269.500)
    CS46        s1a         0          capmod    (1562.650,   57.000,    41.000)
    CS47        s2a         0          capmod    (1434.550,   56.250,    40.500)
    CS48        dx          0          capmod    (21442.475,  611.250,   412.000)
    CU49        s7b         0          capmod    (12008.963,  405.750,   268.000)
    CU50        s5b         0          capmod    (4213.375,   69.750,    49.500)
    CU51        ey          0          capmod    (21593.000,  597.750,   401.000)
    CU52        s6a         0          capmod    (3422.500,   60.000,    43.000)
    CW53        qx          0          capmod    (21743.162,  597.000,   401.000)
    CW54        ckyn        0          capmod    (24097.201,  627.000,   422.000)
    CW55        s16b        0          capmod    (6993.000,   153.750,   105.500)
    CW56        qxn         0          capmod    (19240.500,  597.000,   401.000)
    CW57        s7a         0          capmod    (7141.600,   289.500,   194.500)
    CW58        eyn         0          capmod    (23765.350,  594.375,   399.500)
    CW59        dxn         0          capmod    (22213.674,  597.375,   401.500)
*
 .FINIS
*
 .SUBCKT rcr1hd (vbbx,vbby,vcs,ix,ixn:ox,oxn)
*
 .GLOBAL vee,gnd
*
*       SUBCIRCUIT CALLS
*
    X1          (s2,gnd)    p2k4s;* a/scdiff#1/res2d#1/r22
    X2          (s2,gnd)    p2k4s;* a/scdiff#1/res2d#1/r21
    X5          (s3,gnd)    p2k4s;* a/scdiff#1/res2d#1/r18
    X6          (s3,gnd)    p2k4s;* a/scdiff#1/res2d#1/r17
    X17         (vee,s4)    p1p51k4;* a/scdiff#1/res1a#1/r14
    X19         (vee,s4)    p1p51k4;* a/scdiff#1/res1a#1/r13
    X25         (s1,vcs,s4)   e1xr;* a/scdiff#1/q5
    X33         (gnd,s2,ox)   e1xr;* a/scdiff#1/q21
    X35         (s2,ixn,s1)   e1xr;* a/scdiff#1/q16
    X37         (gnd,s3,oxn)   e1xr;* a/scdiff#1/q20
    X39         (s3,ix,s1)   e1xr;* a/scdiff#1/q14
*
*       CAP/DIODE CARDS
*
*
    CA1         s1          0          capmod    (6065.150,   150.500,   154.500)
    CG2         s4          0          capmod    (4335.350,   37.000,    33.000)
    CI3         ixn         0          capmod    (1372.088,   23.625,    20.500)
    CI4         ix          0          capmod    (1372.088,   23.625,    20.500)
    CI5         oxn         0          capmod    (2733.450,   41.500,    45.000)
    CK6         ox          0          capmod    (2733.450,   41.500,    45.000)
    CK7         s2          0          capmod    (8391.550,   143.500,   137.500)
    CK8         s3          0          capmod    (8391.550,   143.500,   137.500)
*
*
    CS9         ix          0          capmod    (8879.275,   232.500,   159.500)
    CU10        ox          0          capmod    (18617.100,  597.000,   401.000)
```

```
   CU11        ixn       0         capmod    (8879.275,   232.500,   159.500)
   CW12        oxn       0         capmod    (18617.100,   597.000,   401.000)
*
 .FINIS
*
 .SUBCKT x1mn (vbbx,vbby,vcs,ixp,ixn:on)
*
 .GLOBAL vee,gnd
*
*       SUBCIRCUIT CALLS
*
   X10       (s3,gnd)   p0p35k12;* a/res4b#1/r18
*   X11       (s3,gnd)   p0p35k12;* a/res4b#1/r17
   X12       (s2a,vee)   p0p09k24;* a/res3b#1/r19
   X13       (s2a,s2)   p0p09k24;* a/res3b#1/r20
   X44       (s1,vcs,s2)   e8xr;* a/q18
   X45       (gnd,ixp,s1)   e13xr;* a/q17
   X46       (s3,ixn,s1)   e13xr;* a/q16
   X49       (gnd,s3,on)   e53x;* a/q15
*
*       CAP/DIODE CARDS
*
*
   CG1         s2a       0         capmod    (60556.801,   456.000,    86.000)
   CI2         ixn       0         capmod    (3196.550,    51.625,    36.500)
   CK3         s2        0         capmod    (13453.691,   138.500,    77.250)
   CK4         on        0         capmod    (137363.266,  2385.875,   387.750)
   CK5         s1        0         capmod    (11065.601,   197.000,   106.500)
   CK6         ixp       0         capmod    (4616.400,    67.375,    45.500)
   CK7         s3        0         capmod    (27449.145,   481.000,   323.500)
*
*
   CS8         on        0         capmod    (16235.300,   451.750,   152.000)
   CU9         ixn       0         capmod    (16260.894,   336.000,   228.500)
   CU10        ixp       0         capmod    (17122.961,   343.500,   233.500)
*
 .FINIS
*
 .SUBCKT xbmb (vbbx,vbby,vcs,ixp,ixn:on,oi)
*
 .GLOBAL vee,gnd
*
*       SUBCIRCUIT CALLS
*
   X10       (s3,gnd)   p0p35k12;* b/res4b#1/r18
   X11       (s3,gnd)   p0p35k12;* b/res4b#1/r17
   X14       (s3,gnd)   p2k4s;* b/eclin#1/res2db#1/r16
   X15       (s3,gnd)   p2k4s;* b/eclin#1/res2db#1/r15
   X16       (s3,gnd)   p2k4s;* b/eclin#1/res2db#1/r14
   X17       (s3,gnd)   p2k4s;* b/eclin#1/res2db#1/r13
   X24       (vee,s6a)   p1p51k4;* b/eclin#1/res1b#1/r4
   X27       (s6a,s6)   p1p51k4;* b/eclin#1/res1b#1/r10
   X28       (s6a,s6)   p1p51k4;* b/eclin#1/res1b#1/r9
   X29       (vee,s6a)   p1p51k4;* b/eclin#1/res1b#1/r3
   X45       (gnd,ixn,s6)   e13xr;* b/q17
   X46       (s3,s6,s1)   e13xr;* b/q16
   X49       (gnd,s3,on)   e53x;* b/q15
   X59       (s4,gnd)   p0p35k12;* a/res4b#1/r18
   X60       (s4,gnd)   p0p35k12;* a/res4b#1/r17
   X61       (vee,s2)   p0p09k24;* a/res3b#1/r19
   X62       (vee,s2)   p0p09k24;* a/res3b#1/r20
   X63       (s4,gnd)   p2k4s;* a/eclin#1/res2db#1/r16
   X64       (s4,gnd)   p2k4s;* a/eclin#1/res2db#1/r15
   X65       (s4,gnd)   p2k4s;* a/eclin#1/res2db#1/r14
```

```
X66       (s4,gnd)  p2k4s;* a/eclin#1/res2db#1/r13
X73       (vee,s5a)  p1p51k4;* a/eclin#1/res1b#1/r4
X76       (s5a,s5)  p1p51k4;* a/eclin#1/res1b#1/r10
X77       (s5a,s5)  p1p51k4;* a/eclin#1/res1b#1/r9
X78       (vee,s5a)  p1p51k4;* a/eclin#1/res1b#1/r3
X93       (s1,vcs,s2)  e8xr;* a/q18
X94       (gnd,ixp,s5)  e13xr;* a/q17
X95       (s4,s5,s1)  e13xr;* a/q16
X98       (gnd,s4,oi)  e53x;* a/q15
*
*       CAP/DIODE CARDS
*
*
CE1         s5a         0         capmod    (13372.500,  100.000,   40.000)
CG2         s5          0         capmod    (14252.562,  172.500,  130.000)
CG3         s2          0         capmod    (25045.977,  277.000,  146.500)
CG4         s6a         0         capmod    (13372.500,  100.000,   40.000)
CI5         s6          0         capmod    (14252.563,  172.500,  130.000)
CI6         ixp         0         capmod    (4006.275,   63.000,    43.000)
CI7         ixn         0         capmod    (4006.275,   63.000,    43.000)
CK8         s4          0         capmod    (35332.883,  596.375,  391.500)
CK9         oi          0         capmod    (137363.266, 2385.875,  387.750)
CK10        on          0         capmod    (137363.266, 2385.875,  387.750)
CK11        s3          0         capmod    (35332.883,  596.375,  391.500)
CK12        s1          0         capmod    (22032.549,  378.500,  188.250)
*
*
CQ13        s5          0         capmod    (20614.600,  477.000,  282.000)
CS14        oi          0         capmod    (16235.300,  451.750,  152.000)
CS15        on          0         capmod    (16235.300,  451.750,  152.000)
CU16        s4          0         capmod    (15026.625,  310.500,  200.000)
CU17        ixn         0         capmod    (9156.700,  255.750,  175.000)
CW18        s6          0         capmod    (20614.598,  477.000,  282.000)
CW19        ixp         0         capmod    (9788.150,  255.750,  175.000)
CW20        s3          0         capmod    (15026.625,  310.500,  200.000)
*
 .FINIS
```

12. Vita

Donald J. Wingate was born on Johnston Air Station in Saitama Ken, Japan on September 25, 1962 to Dale and Joan Wingate. He graduated from Eleanor Roosevelt Senior High School in Greenbelt, Maryland in June, 1980. He continued his academic career at The Pennsylvania State University, receiving a Bachelors of Science degree in Electrical Engineering in June 1984. In June 1985 Donald Wingate married Candy Willis, and four years later in December 1989 were graced with a beautiful daughter, Brandi Noel.

The Wingates have resided in Northampton, Pennsylvania since 1987 and Donald Wingate has been employed by AT&T Microelectronics, Inc. since June 1984, where he is a designer of digital bipolar integrated circuits.