CSG-30 JULY, 1984

COORDINATED SCIENCE LABORATORY
COMPUTER SYSTEMS GROUP

DESIGN OF A MICROPROGRAM CONTROL UNIT WITH CONCURRENT ERROR DETECTION

MARY M. YEN

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS N/A
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CSG-30	5. MONITORING ORGANIZATION REPORT NUMBER(S) N/A
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Laboratory University of Illinois 6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION Office of Naval Research
6c. ADDRESS (City, State and ZIP Code) 1101 West Springfield Avenue Urbana, IL 61801	7b. ADDRESS (City, State and ZIP Code) 2511 Jefferson Davis Highway Arlington, VA 22202
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 8b. OFFICE SYMBOL (If applicable) N/A	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00039-80-C-0556
8c. ADDRESS (City, State and ZIP Code) 2511 Jefferson Davis Highway Arlington, VA 22202	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: N/A PROJECT NO.: N/A TASK NO.: N/A WORK UNIT NO.: N/A
11. TITLE (Include Security Classification) Design of a Microprogram Control Unit with Concurrent Error Detection	
12. PERSONAL AUTHOR(S) Yen, Mary M.	
13a. TYPE OF REPORT Technical 13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) August 1984 15. PAGE COUNT 44
16. SUPPLEMENTARY NOTATION N/A	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Concurrent Error Detection, Fault Tolerance, Microprogram Control Unit, Strongly Code Disjoint, Strongly Fault Secured, Totally Self Checking, VLSI

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper presents an integrated approach to the design of a microprogram control unit (MCU) with concurrent error detection (CED) capability for errors generated by VLSI physical failures. The paper first presents the design of a single-chip MCU that comprehensively detects errors due to internal physical failures during its normal operation. The AM2910 microprogram sequencer is used as a functional model for the CED MCU. Lastly, the paper presents a critical evaluation of the actual mask-level layout of the CED MCU design versus a simplex MCU without CED and a CED MCU through duplication and comparison.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code) 22c. OFFICE SYMBOL N/A

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

ACKNOWLEDGMENT

The author wishes to express appreciation to her thesis advisor, Professor J. A. Abraham, and Professor E. S. Davidson for starting her on the thesis research area. The author also wants to express special gratitude to W. K. Fuchs for his suggestions and support. Finally, Special thanks go to Joe Rahmeh and Bill Rogers for their help in organizing the layout and simulation tools.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Because of greater reliability demands placed upon the modern digital systems, these systems need to be designed with fault-tolerant capability. Concurrent error detection (CED) can provide this capability by detecting errors caused by faults in the system during normal operation of the system. Also with CED, an error can be detected soon after it is produced, resulting in shorter error latency and easier error recovery. One application of CED is on a microprogram control unit (MCU).

Much research has been done in the area of CED, including coding and self-checking circuits [Wake78] and time redundancy [PaFu82]. However, the CED concept is mainly applied to various codes, data transmission, and simple functional units, such as arithmetic units. Little work has been done in the control unit area. Previous work is primarily in the use of classical self-checking circuits, using bit slicing, parity, and m-out-of-n codes in simple control units to detect a limited class of faults [CSST73], [DiSo75], [Maki78], [Will77]. These techniques are neither applicable to a complex control unit, like the AM2910, nor to the VLSI technology.

The only proposals applicable to the above two constraints have been self-checking MOS-LSI circuits using coding [CrLa80] and duplication [Wake78], [SeLi80]. In [CrLa80], the self-checking technique is applied to a microprocessor; however, the design is not an actual chip design. Comparisons are done in terms of number of transistors and not in terms of actual chip area. The duplication technique requires not only duplicated control units but also input and output checkers and an output check bit generator. The area redundancy of the duplication technique will be compared in Chapter 6 to the design introduced in this thesis.

Recent research in the control unit area has proposed methods using a parallel signature analyzer [Namj82], [DuMa83], a check symbol stored in the control memory [IyKi82], or a separate watchdog monitor [SrTh82]. The signature error detection scheme is based on percentage of error detection but not on any fault model, and the scheme does not detect incorrect branches. The check symbol scheme does not detect all illegal and incorrect branches and does not have a comprehensive bit error detection. The performance of the watchdog monitor scheme is unclear because it depends on the complexity of the monitor.

All of the above proposals in the CED area are not based on actual chip layout. There are only two proposals based on actual chip layout: the C.fast chip [TWMTS82] and the MCU chip [WFAD83]. The C.fast chip is a single chip fault-tolerant microprocessor. The C.fast chip uses simple PLAs with parity checking as its controller. There is no protection for portions of the chip, such as the control bus and the ALU. Also, the retry PLA is not implemented on the chip. The MCU chip is a microsequencer, based on the AM2910, with CED. This thesis is on the redesign and layout of the MCU chip.

Chapter 2 gives a functional description of AM2910 upon which our design is based. Some modifications have been made for CED and technology considerations, and these modifications are discussed. The resultant modified instruction set is also given.

Chapter 3 develops a fault model for the MCU. Instead of considering every possible physical fault on the MCU, the functional level fault model developed in [BaAb82] is used. Six potential areas for error are discussed.

In Chapter 4, modifications made on Wong's design are discussed. All modifications are classified into four levels: system, layout, performance, and area. At the system level, changes are made to improve the CED fault coverage. Some modifications are made at the layout level due to process changes. At the performance level, the main emphases are to minimize delay time and to decrease the clock cycle. Finally, at the area level, redundancy is kept to a minimum.

Chapter 5 begins with an overview of the CED design approach and is continued with a detailed CED design on the MCU. Individual functional modules and checkers are discussed.

Chapter 6 is devoted to evaluation of the chip design in terms of area redundancy and timing performance. For timing evaluation, TSIM, a MOS timing simulator, is used on all modules. Based on TSIM results, critical paths are found for the MCU. Redundancy and performance of the MCU are compared to the Wong's design and also to the duplication approach.

Chapter 7 provides conclusions and suggestions for further research. Finally, the appendix contains figures for various cell design in mixed notation.

CHAPTER 2

THE MICROPROGRAM SEQUENCER

2.1. The AM2910

The AM2910 Microprogram controller is a 12-bit bipolar address sequencer for up to 4K words of microprogram, as shown in Figure 2-1. During each microinstruction, the multiplexer selects an address (Y) from one of four sources: register/counter (R/C), microprogram counter (UPC), stack or direct external input (X). The instruction programmable logic array (PLA) decodes 4-bit instruction input (I) into internal control signals. The output of the PLA is affected by the condition code (CC) and zero-detection (R=0) signal from the R/C.

2.2. Modifications

Several modifications have been made to account for nMOS technology and CED consideration, as shown in Figure 2-2. A two-phase clock (PHI1 and PHI2) is used. Instruction execution and error checking are pipelined. During PHI1, the instruction is decoded, then during PHI2, the output address Y is generated. During the next clock cycle, the next instruction is decoded in PHI1, and the status signals of the previous instruction are generated in PHI2. Detailed timing operations are discussed in Section 6.1.

Several simplifications have also been made. Condition code enable $\overline{\text{CCEN}}$ has been omitted. The three enable signals (PL, MAP, and VECT) are not in their complemented value as in the AM2910. The register load signal $\overline{\text{RLD}}$ is also omitted; therefore, R/C can be loaded only by instructions. The UPC is incremented at every cycle, thus eliminating the

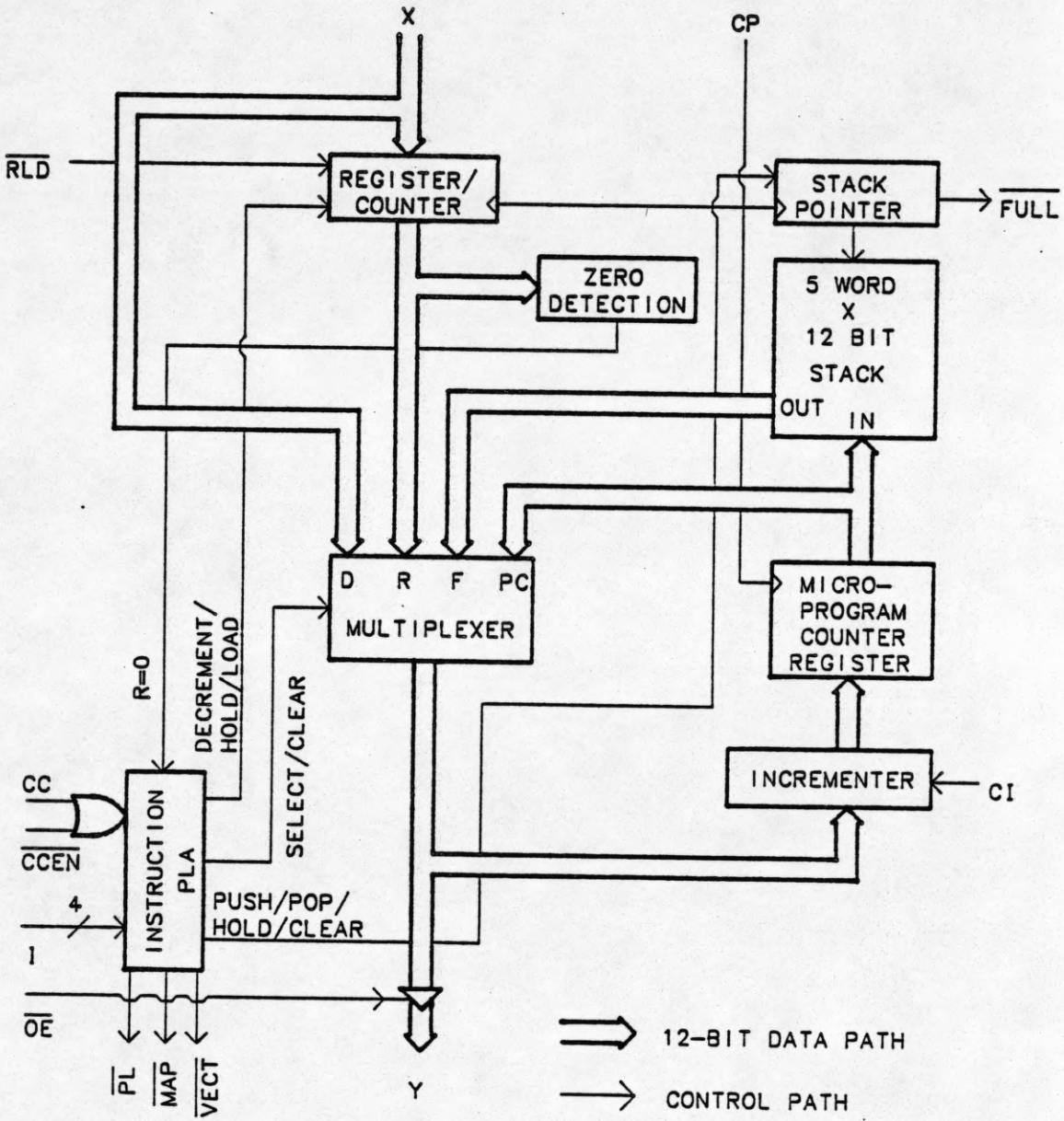

Figure 2-1. AM2910 Block Diagram.

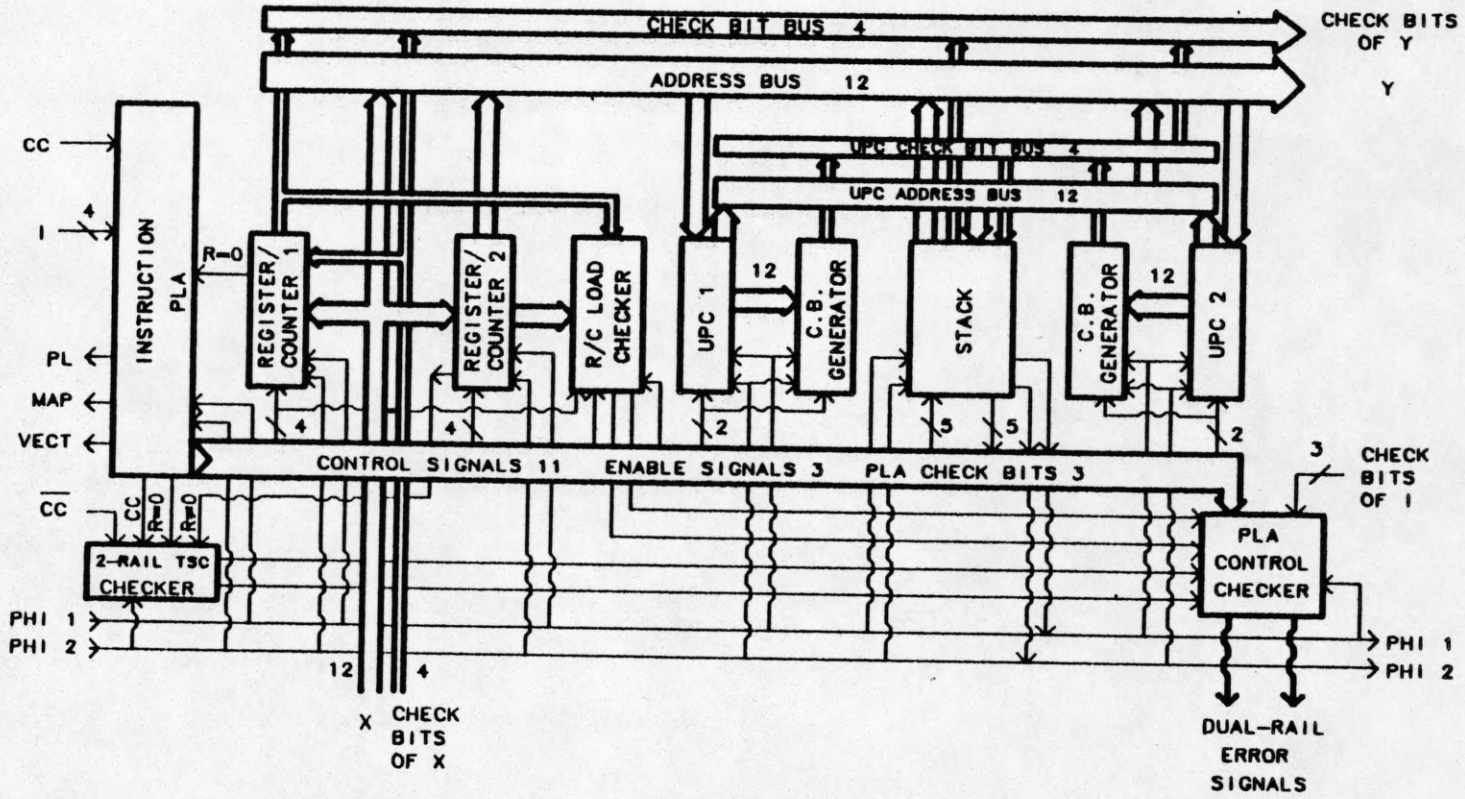

Figure 2-2. MCU Block Diagram.

carry-in (CI) input. The omission of CI does not allow the MCU to operate as a slice of a multichip MCU, as the case of the AM2910. The Y output is always enabled so that output enable $\overline{OE}$ is eliminated. The stack $\overline{FULL}$ signal is omitted.

2.3. The Instruction Set

The instruction set after the above modifications is shown in Table 2-1. The instruction set is very similar to the AM2910 instruction set [MiBr80]. The major change is the elimination of $\overline{CCEN}$ For the JUMP ZERO or RESET instruction, the address Y is set to 0 by setting all outputs of the UPC to 0.

Table 2-1. The Instruction Set.

HEX I3-I0	MNE-MONIC	NAME	R/C CON-TENTS	FAIL CC=LOW		PASS CC=HIGH		R/C	EN-ABLE
				Y	STACK	Y	STACK		
0	JZ	JUMP ZERO	X*	UPC	HOLD	UPC	HOLD	HOLD	PL
1	CJS	COND JSB PL	X	UPC	HOLD	EXT	PUSH	HOLD	PL
2	JMAP	JUMP MAP	X	EXT	HOLD	EXT	HOLD	HOLD	MAP
3	CJP	COND JUMP PL	X	UPC	HOLD	EXT	HOLD	HOLD	PL
4	PUSH	PUSH/COND LD CNTR	X	UPC	PUSH	UPC	PUSH	**	PL
5	JSRP	COND JSB R/PL	X	REG	PUSH	EXT	PUSH	HOLD	PL
6	CJV	COND JUMP VECTOR	X	UPC	HOLD	EXT	HOLD	HOLD	VECT
7	JRP	COND JUMP R/PL	X	REG	HOLD	EXT	HOLD	HOLD	PL
8	RFCT	REPEAT LOOP, CNTR ≠ 0	≠ 0	STACK	HOLD	STACK	HOLD	DEC	PL
			= 0	UPC	POP	UPC	POP	HOLD	PL
9	RPCT	REPEAT PL, CNTR ≠ 0	≠ 0	EXT	HOLD	EXT	HOLD	DEC	PL
			= 0	UPC	HOLD	UPC	HOLD	HOLD	PL
A	CRTN	COND RETURN	X	UPC	HOLD	STACK	POP	HOLD	PL
B	CJPP	COND JUMP PL & POP	X	UPC	HOLD	EXT	POP	HOLD	PL
C	LDCT	LD CNTR & CONTINUE	X	UPC	HOLD	UPC	HOLD	LOAD	PL
D	LOOP	TEST END LOOP	X	STACK	HOLD	UPC	POP	HOLD	PL
E	CONT	CONTINUE	X	UPC	HOLD	UPC	HOLD	HOLD	PL
F	TWB	THREE WAY BRANCH	≠ 0	STACK	HOLD	UPC	POP	DEC	PL
			= 0	EXT	POP	UPC	POP	HOLD	PL

* X = Don't care.

** If fail, HOLD, else LOAD.

CHAPTER 3

FAULT MODEL

3.1. Functional Fault Model

Before designing CED capability onto the MCU, a set of faults must be predefined so that CED will detect errors caused by these faults. When the chip is as complex as the MCU, the classical stuck-at fault model is insufficient to describe all possible faults on the chip

Instead of defining faults on single lines, faults can be classified at the functional level [BaAb82]. A module can be divided into functional blocks: PLA, decrementer, incrementer, register, etc. Each block is described by the functional effects of the physical faults on the function of the block. Based on the functional fault model approach, a fault model is developed for the MCU.

3.2. Fault Model for the MCU

The MCU has six potential areas for error:

(1) Input controls signals (I, CC).

(2) External inputs (X).

(3) Control decoding and transferring.

(4) Modules (decrementer, incrementer, and stack).

(5) Address Bus.

(6) Power.

The first two areas include errors occurring during signal transmission. The third area includes errors in the instruction PLA and the PLA control bus. A single physical failure in PLA will cause unidirectional errors at the output [BaAb82]. Faults in the control bus can cause misselection: selecting the wrong source, selecting two sources, or no selection. Selection of two sources will result in unidirectional errors that can be detected on the address bus. When no source is selected, all 1s will appear on the address bus. The fourth area includes not only errors in the R/C, UPC, and stack but also errors in the fanout lines of the PLA control signals. Because errors resulting from faults on the R/C and UPC are not clear, random errors are assumed. The fifth area covers all bus errors. Bridging faults or broken bit bus lines cause unidirectional error in nMOS technology. The final area is on power failure in the major fanout of power and ground lines, which will cause those nodes to be floating.

CHAPTER 4

CHANGES FROM WONG'S DESIGN

This MCU design has many changes from Wong's design [WFAD83]. Detailed information on Wong's design is available in [Wong82]. All the changes can be classified into four levels: system, area, performance, and layout.

At the system level, changes are made to simplify the design without diminishing the CED capability. First, the address checker has been eliminated, which is made possible by checking the output of the MCU along with the output of the micro-store using a CED scheme proposed in [FuAb84]. The same scheme is used for the PLA and PLA control checker; similarly, the PLA input checker is eliminated. To improve the fault coverage of the MCU, both the UPC and its check-bit generator are duplicated, and a checker is added for checking R/C against its check bits when loaded with external inputs.

At the layout level, three changes are made. The first is the change from the Texas Instruments design rules to Mead and Conway design rules [MeCo80]. Because of processing requirements, buried contact is used instead of butting contact, and the value of lambda width is changed from 2.5 microns to 2 microns.

At the area level, the effort is to minimize area redundancy. A check-bit generator is shared by both the R/C load checker and the PLA control checker. Two-rail totally self-checking checkers are replaced by TSC checkers, proposed by [JhAb84], because the latter requires less area than the former. The elimination of the address checker, input checker, and register tags at the system level, as mentioned before, also result in reduction of area redundancy.

At the performance level, the overall cycle time is reduced by pipelining the instruction execution and checking. Also, many of the basic cells, such as adders and subtractors, are redesigned to have shorter delay time by using a pass transistor networks [Whit83].

CHAPTER 5

THE DESIGN OF THE CED MCU

5.1. An Overview of the CED

All information is encoded with a Berger code, which is the binary count of the number of zeros in the information. The Berger code is selected because it is a systematic code, where the information bits are separated from the code bits and because the code can detect all unidirectional errors in a code word.

All input signals are checked within the chip. Instruction signals (I) and external input signals (X) are encoded with Berger code, as shown in Figure 2-2. Both CC and $\overline{CC}$ are input for two-rail checking.

The output address is encoded for off-chip checking. Three enable signals, pipeline address enable (PL), map address enable (MAP), and vector address enable (VECT), are output from the MCU. These enable signals select the source for direct input source. Since only one of the three signals is HIGH at any time, the three enable signals form a 1-out-of-3 code for off-chip checking. The two clock signals are output from the chip to detect any error in the clock signals.

A strongly fault secured and strongly code disjoint PLA is used [FuAb84]. A modified Berger code is used over both the outputs and the inputs (I). The register/counter and UPC are duplicated to detect random errors. The stack is a strongly fault secure shift stack. The strongly fault secure multiplexer takes on a bus structure. As mentioned in Chapter 4, the checking of the address bus has been moved off-chip.

Two totally self-checking checkers are used. The first one is the R/C load checker. When the R/C is loaded with external inputs, its register content is checked against its

Berger check bits. The checking is necessary to insure that the value, if used for counting, is correct.

The second checker is the PLA control checker. This checker provides error detection in the following areas: input control signals, PLA decoding, and control signal transferring. It also provides TSC capability to the stack and to the multiplexer by placing it at the end of the control bus, after the control signals have passed through various modules.

The power and clock signals take on bus structures. The signals come into the chip from one end and routed to the other end of the chip through bus lines. The PLA control checker is placed at the end of the power bus to detection power failure. The two clock phases are output from the chip at the end of the clock bus.

1.1. Functional Description

The PLA has six inputs: 4-bit instruction input (I), condition code (CC), and register-zero-detection (R=0). The zero-detection is an internal input. The PLA generates nine internal control signals, two of which are also inverted at the PLA output. Besides the control signals, the PLA also produces three enable signals: PL, MAP, and VECT.

The PLA is encoded in a modified Berger code [MaAD82]. As shown in Table 5-1, the number of zeros in both input instruction (I) and 12-bit output is from 8 to 14. The modified Berger code requires 3 bits to encode 0 to 6 for 8 to 14 zeros. Counting the 3-bit code word, the PLA generates a total of 17 outputs.

The R/C is used either as a register to hold a branch address or as a loop counter by decrementing the content of the register. When the external input is loaded into R/C, the information is checked against the check bits by the R/C load checker. Once the register has been decremented, the register should not be selected as the source of the multiplexer. During PHI2, R/C 1 generates R=0 signal for the PLA, while R/C 2 generates R≠0 for two-rail checking.

Table 5-1. PLA Input and Output Patterns.

Hex I3-I0	CC	R/C R=0	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0	No. of 0s *	CB2	CB1	CB0
0	X	X	1	1	0	0	0	0	0	0	0	1	0	0	13	1	0	1
1	1	X	0	0	0	0	0	1	0	0	1	1	0	0	12	1	0	0
	0	X	0	1	0	0	0	0	0	0	0	1	0	0	13	1	0	1
2	X	X	0	0	0	0	0	1	0	0	0	0	1	0	13	1	0	1
3	1	X	0	0	0	0	0	1	0	0	0	1	0	0	12	1	0	0
	0	X	0	1	0	0	0	0	0	0	0	1	0	0	12	1	0	0
4	1	X	0	1	1	0	0	0	0	0	1	1	0	0	11	0	1	1
	0	X	0	1	0	0	0	0	0	0	1	1	0	0	12	1	0	0
5	1	X	0	0	0	0	0	1	0	0	1	1	0	0	11	0	1	1
	0	X	0	0	0	0	1	0	0	0	1	1	0	0	11	0	1	1
6	1	X	0	0	0	0	0	1	0	0	0	0	0	1	12	1	0	0
	0	X	0	1	0	0	0	0	0	0	0	0	0	1	12	1	0	0
7	1	X	0	0	0	0	0	1	0	0	0	1	0	0	11	0	1	1
	0	X	0	0	0	0	1	0	0	0	0	1	0	0	11	0	1	1
8	X	1	0	1	0	0	0	0	0	1	0	1	0	0	12	1	0	0
	X	0	0	0	0	1	0	0	1	0	0	1	0	0	12	1	0	0
9	X	1	0	1	0	0	0	0	0	0	0	1	0	0	12	1	0	0
	X	0	0	0	0	1	0	1	0	0	0	1	0	0	11	0	1	1
A	1	X	0	0	0	0	0	0	1	1	0	1	0	0	11	0	1	1
	0	X	0	1	0	0	0	0	0	0	0	1	0	0	12	1	0	0
B	1	X	0	0	0	0	0	1	0	1	0	1	0	0	10	0	1	0
	0	X	0	1	0	0	0	0	0	0	0	1	0	0	11	0	1	1
C	X	X	0	1	1	0	0	0	0	0	0	1	0	0	11	0	1	1
D	1	X	0	1	0	0	0	0	0	1	0	1	0	0	10	0	1	0
	0	X	0	0	0	0	0	0	1	0	0	1	0	0	11	0	1	1
E	X	X	0	1	0	0	0	0	0	0	0	1	0	0	11	0	1	1
F	1	1	0	1	0	0	0	0	0	1	0	1	0	0	9	0	0	1
		0	0	1	0	1	0	0	0	1	0	1	0	0	8	0	0	0
	0	1	0	0	0	0	0	1	0	1	0	1	0	0	9	0	0	1
		0	0	0	0	1	0	0	1	0	0	1	0	0	9	0	0	1

F11 = Reset.
F10 = UPC output enable.
F9 = R/C load.
F8 = R/C decrement.
F7 = R/C output enable.
F6 = External address output enable.
F5 = Top of stack output enable.
F4 = Stack POP.
F3 = Stack PUSH.
F2 = Pipeline register enable (PL).
F1 = Map PROM enable (MAP).
F0 = Vector register enable (VECT).
X = Don't care.
* Number of 0s in I3-I0 and F11-F0.

The UPC increments the current address at each clock cycle and generates the check bits for the incremented address. When the RESET instruction (instruction 0) is executed, the output of the UPC is set to address 0 and the output of the check-bit generator is set to the corresponding Berger code. The UPC and its check-bit generator are both duplicated. The outputs of the duplicated modules are hardwired AND together as shown is Figure 5-1. If any one of the copy is faulty, unidirectional errors are resulted in the ANDed output, which is detectable by the Berger code.

The 5-word by 16-bit last-in, first-out stack provides return address for microsubroutines or loops. The stack is a modified shift stack in [MeCo80]. The stack is PUSHed during PHI1 from the UPC bus and the check-bit bus, and is POPed during PHI2 unto the address bus. Both information and check bits are stored in the stack. The stack is made to be TSC by checking the control signals after they passed through the stack.

The address bus, the output of the multiplexer, is precharged during PHI1. During PHI2, one of the four possible inputs is enabled onto the address bus. The multiplexer is made to be TSC by checking the enable control signals after they pass through the multiplexer.

The totally self-checking checker consists of a check-bit generator and a totally self-checking equality checker. The check-bit generator is a counter using full adders and half adders connected in a Wallace tree form [WiWi77], as shown in Figure 5-2. The equality checker is built from four-input two-rail TSC checkers in an Anderson tree [Ande71]. Two TSC checkers are used: R/C load checker and PLA control checker.

The R/C load checker, Figure 5-3, operates only when the the R/Cs are loaded. When the LOAD control signal is HIGH, the external input signals (X) are loaded into both R/C 1 and R/C 2, and the check bits of X are loaded only into R/C 1. The check bits from R/C 1 are checked against the check bits generated from the information of the R/C 2. The loaded value is checked to insure that the correct value has been loaded for subsequent decrement.

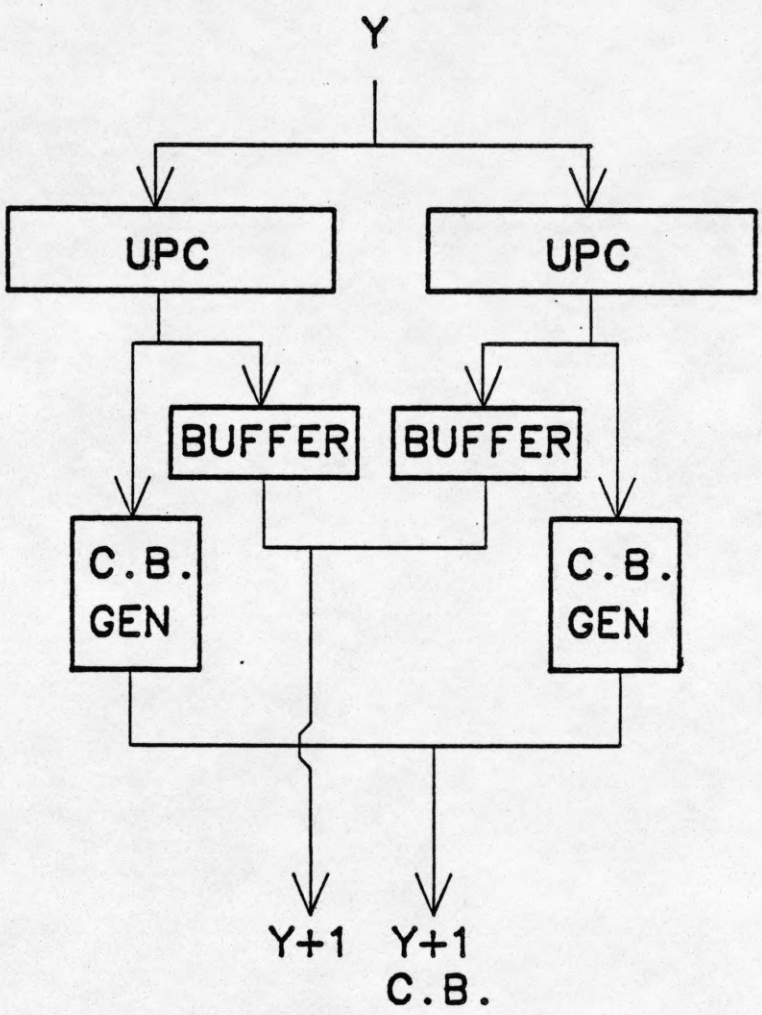

Figure 5-1. UPCs and Check-Bit Generators Block Diagram.

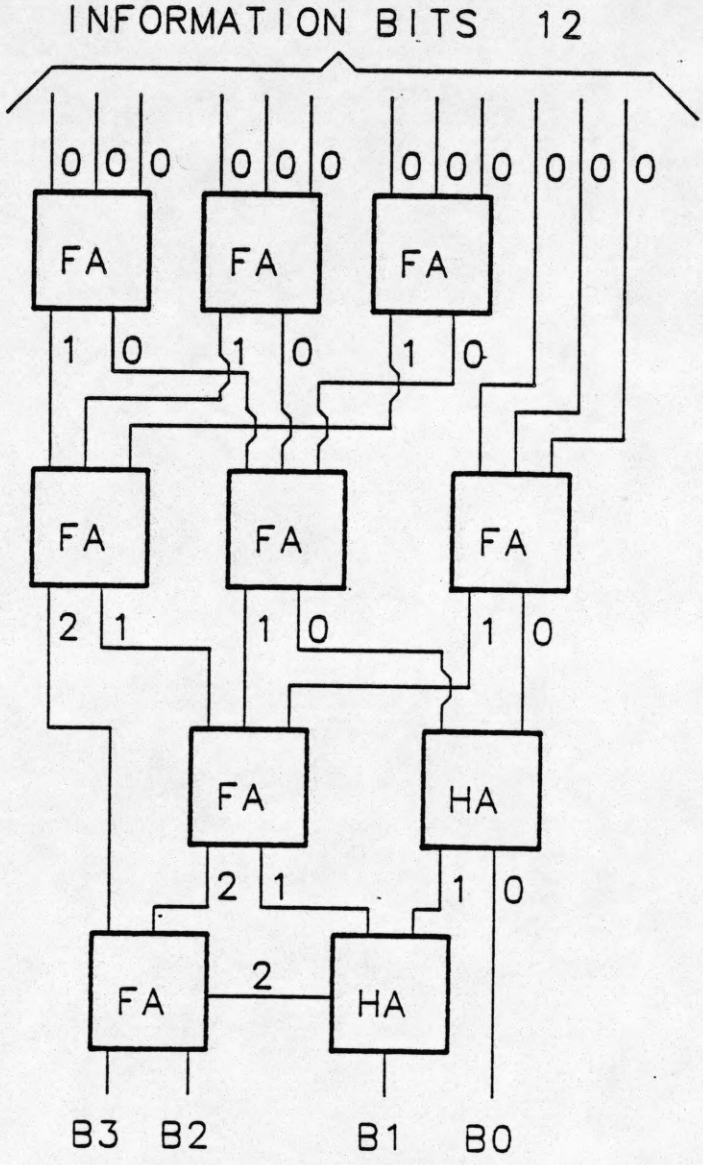

Figure 5-2. Check-Bit Generator.

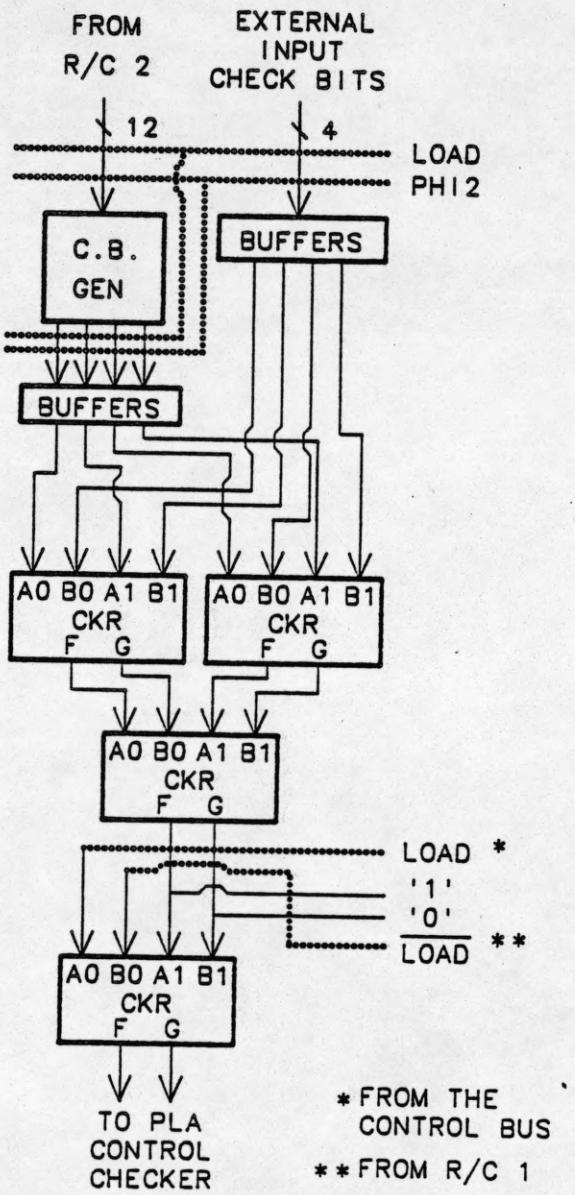

Figure 5-3. Register/Counter Load Checker.

The PLA control checker, Figure 5-4, works in the following way. The check bits of the input control signals (I) are subtracted from the modified Berger code outputs of the PLA. The difference should be the codeword of the 12-bit PLA outputs and is compared with the codeword generated from the PLA output control signals. The other two PLA inputs, CC and R=0, are compared with their $\overline{CC}$ external input and R≠0 from R/C 2, respectively. Two inverted control signals, $\overline{PUSH}$ and $\overline{POP}$, that are not primary outputs of the PLA, are checked against their complements. Furthermore, the output of the R/C load checker is input into the PLA checker. Because of the delay time of the various inputs, the checker is arranged with a minimum amount of delay time.

To have a TSC checker, the checker must have all possible input vectors to exercise all possible faults in the check-bit generator. The PLA control checker cannot meet this requirement because of the specified PLA outputs. This problem can be solved by sharing the check-bit generator between the two checkers. Because there is no restriction on the R/C, all possible input vectors can be produced. Because of the different checking timing, the R/C load checker and the PLA control checker can easily share one check-bit generator without any timing penalty. Since a check-bit generator requires a relatively large chip area, the sharing scheme provides area saving.

5.3. Chip Layout

The floor plan of the MCU is shown in Figure 5-5. The designs for the PLA cells and the input/output pads are described in [HoSe80].

Because of the CED requirement, there are two layout constraints. The first constraint is the control signal fanout lines. Control signals to duplicated modules must be from different fanout lines. If the duplicated modules receive control signals from the same fanout lines, faults on the control lines could cause same errors in both of the modules; therefore, these errors would be undetectable. Control signals to modules that are

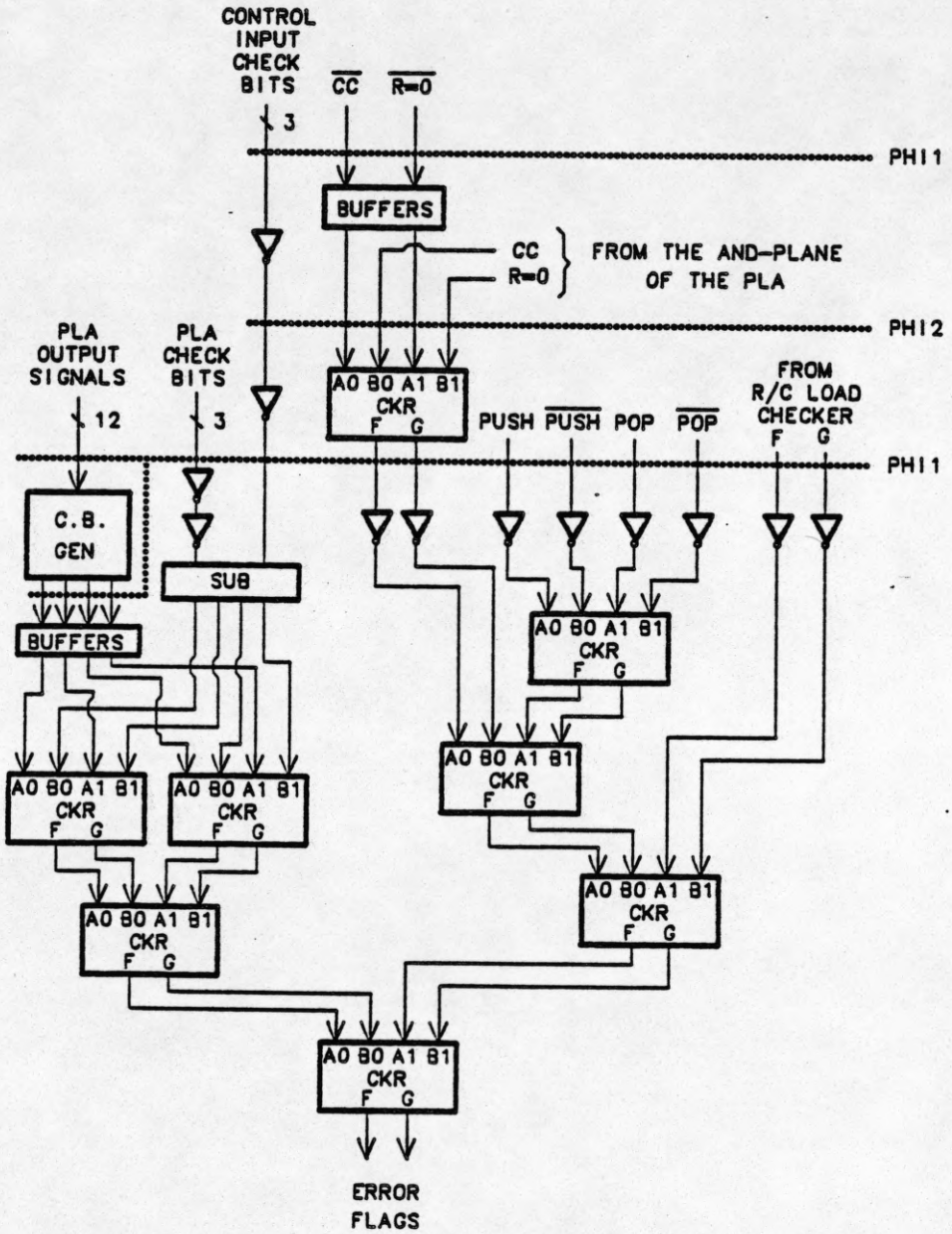

Figure 5-4. PLA Control Checker.

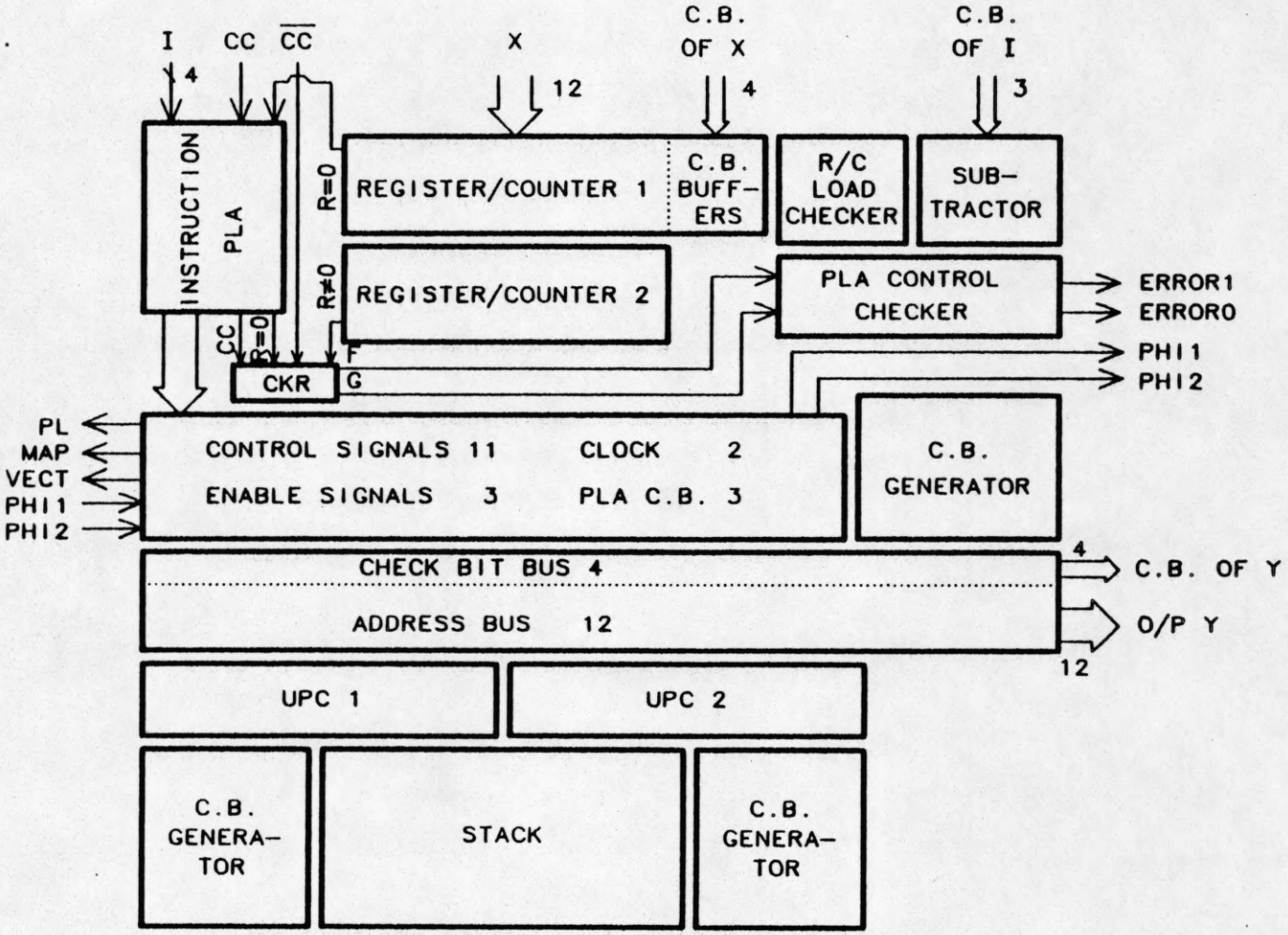

Figure 5-5. Floor Plan.

not duplicated, such as the stack and the multiplexer, are fanout lines from the control bus and are fed back to the control bus. Fanout from the clock and power bus are treated the same way as the control signal fanout by which they are fed back to the original source.

The second constraint is concerning the placement of checkers. The PLA control checker must be placed at the end of the control bus, after all the fanouts and feedbacks. The R/C load checker must be placed to insure at least one of the two R/C copies has the correct value.

CHAPTER 6

EVALUATION AND COMPARISON

6.1. Chip Evaluation

The chip measures 2788 x 2190 microns where lambda = 2 microns in nMOS technology. It contains 4600 transistors and dissipates an estimated 0.24 watts of power with a 5 volt power supply. There are a total of 52 pads: 29 input pads and 23 output pads. A plot of the complete chip layout appears in Figure 6-1.

The area redundancy, due to CED, for the various modules is shown in Table 6-1. The PLA requires no extra AND terms for the check bits, and the three extra outputs account for only 0.7% additional chip area. The redundancy of the R/C contains one copy of the R/C, check-bit buffers, and the bus to the R/C load checker. The redundancy of the UPC includes one copy of the UPC and both copies of the check-bit generator. The redundancy of the stack is in the storing of the check bits. The above three areas also include areas due to control fanout lines. The control bus Both the R/C load checker and the PLA control checker require a total of 19% extra chip area. Because the constraint on the control lines, the control bus must be routed across the chip. The address bus requires redundant area for the check bits. The addition of eight input pads and eight output pads accounts for 14.8% extra area. Because of the placement of the different modules, there are some wasted areas in the layout.

For timing evaluation, TSIM, a MOS timing simulator, is used. Inputs to the simulator are transistor ratios and load capacitances extracted from the layout. Based on simulation, the MCU can be operated with a 300 nanosecond clock cycle. During PHI1, PLA decodes the instruction. During PHI2, the address and its check bits are generated. Internal

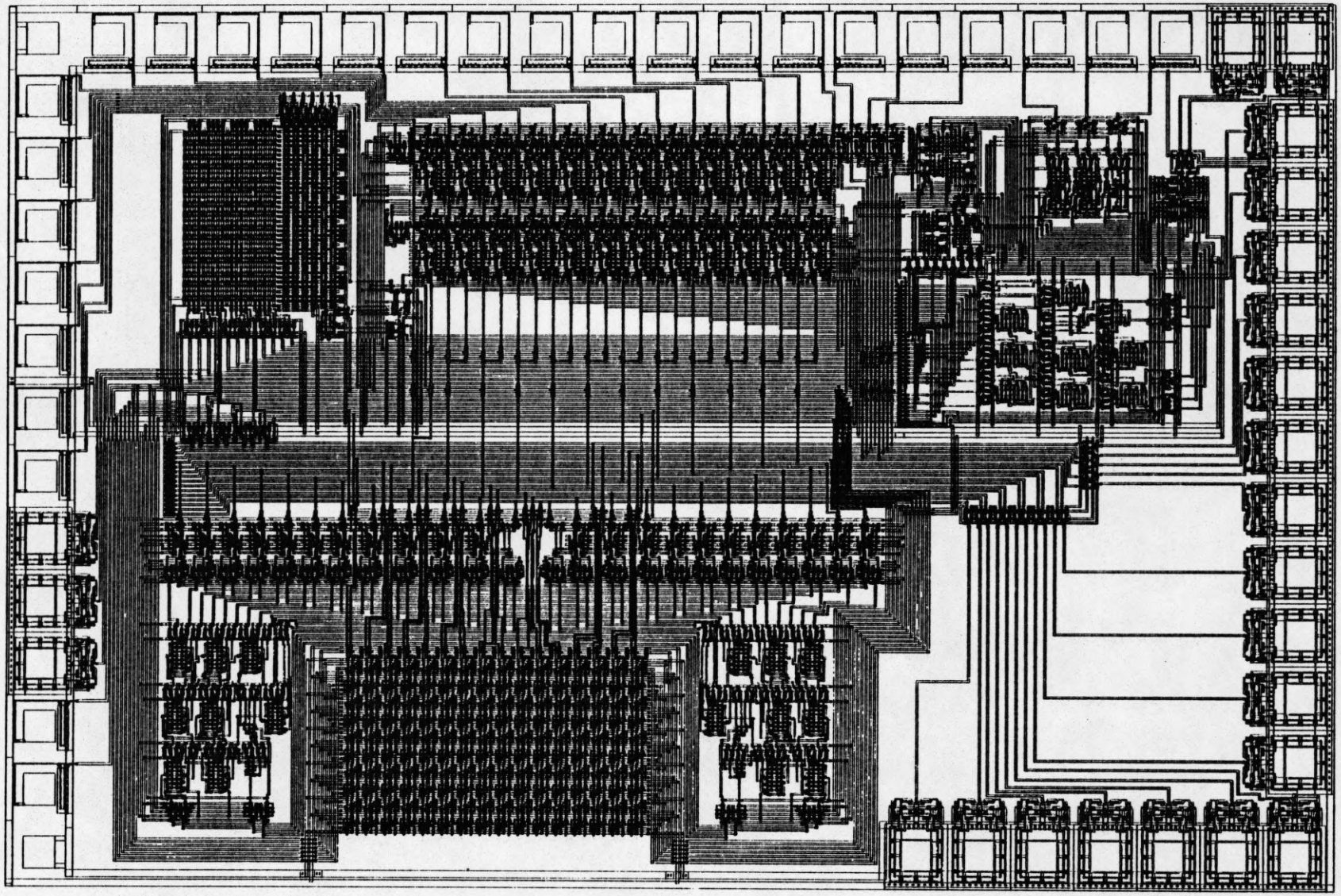

Figure 6-1. Chip Layout Plot.

Table 6-1. MCU Area Redundancy.

	% Area Redundancy
PLA	0.7
RC	13.0
UPC	23.9
Stack	11.3
RC Load and PLA Control Checker	19.0
Control Bus	11.5
Address Bus	10.0
I/O Pads	14.8
Total	104.2

operations start during PHI2, and some are carried into PHI1 of the next clock cycle. The R/C load checker begins checking during PHI2 and sends its 2-bit output to the PLA control checker during PHI1 of the next clock cycle. The PLA control checker starts checking during PHI1 of the next clock cycle, and the status signals become available during PHI2. Based on the above timing operation, the critical path for PHI1 is the decoding of the instruction by the PLA. The critical path for PHI2 is the generation of register-zero (R=0) by the R/C because the R=0 signal is needed for the PLA decoding of the next instruction. The MCU cycle timing waveforms are shown in Figure 6-2.

6.2. Comparison

Since the MCU is based on Wong's design, a comparison is made between the two designs. To evaluate this design approach of the MCU, the MCU is also compared with two other sequencer designs: a simplex sequencer and a single chip sequencer with duplicated control units.

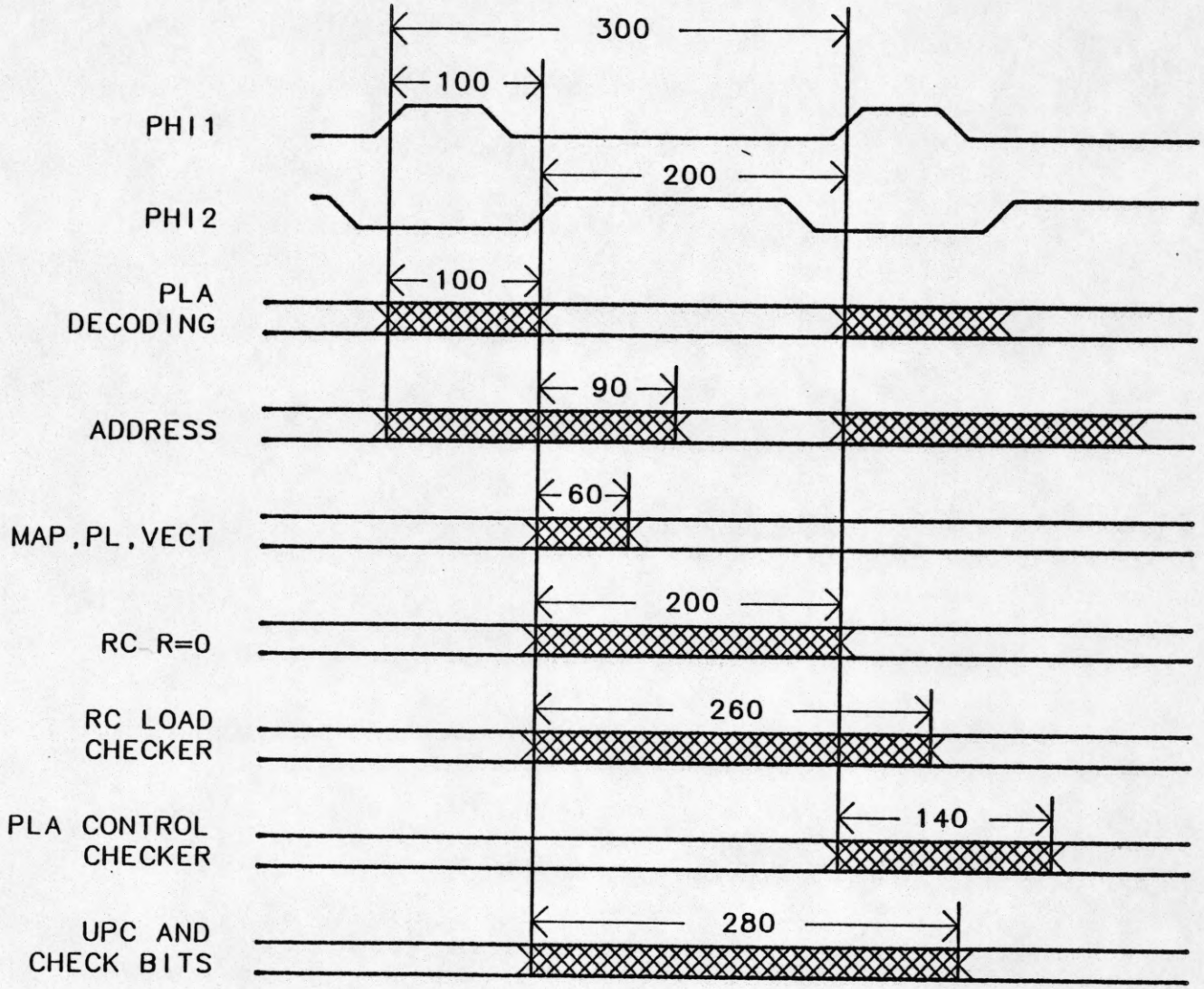

Figure 6-2. MCU Cycle Timing Waveforms.

6.2.1. Comparison to Wong's Design

This design of the MCU has been improved from Wong's MCU (WMCU) both in chip size and in timing performance. The improvement in chip size results from of several factors, as mentioned in Chapter 4. A different set of design rules is used, and lambda is changed from 2.5 microns to 2 microns. Moreover, several function modules are eliminated. The improvement in timing performance can be accounted by the fact that in our design instructions are pipelined. Because of the changes in design rules, lambda width, and design of some basic cells, the delay time of various functional modules has been decreased drastically.

6.2.2. Comparison to a Simple and a Duplicated MCU

This MCU design is compared with two other sequencers: a simplex sequencer and a single chip sequencer with duplicated control units. The simplex sequencer (SMCU) has no checker and the information bits are not encoded. The duplicated sequencer (DMCU), as shown in Figure 6-3, has the same number of input/output pads as the MCU; however, internally it contains duplicated copies of the SMCU without the I/O pads. To provide CED on the DMCU, all input signals must be checked against their check bits; therefore, two input checkers are needed for the instruction and the external address inputs. Also, check bits must be generated for the output address, and an output checker is needed for comparing the outputs from the two copies of the SMCU.

The chip size, timing performance, and power dissipation for the SMCU, MCU, and DMCU are shown in Table 6-2. The area redundancy for the MCU and DMCU are 118% and 138%, respectively. The high redundancy of the MCU can be accounted for by the duplication of the Register/Counter and the UPC. Because of the CED constraint on the control signal lines, a significant part of the redundancy is due to routing. The DMCU has redundancy due to input and output checkers, extra i/o pads, and the complete duplication of the SMCU.

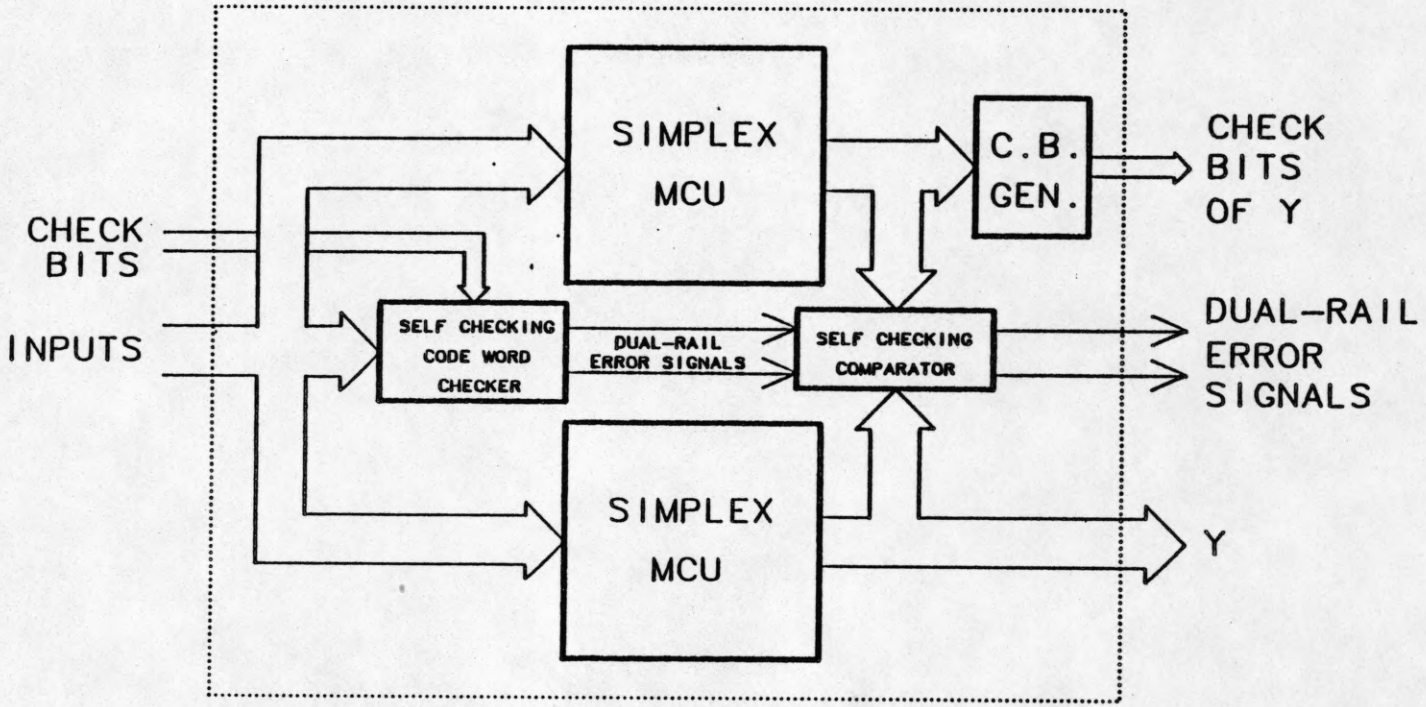

Figure 6-3. Duplicated MCU (DMCU).

Figure 6-3. Duplicated MCU (DMCU).

Table 6-2. Comparison Between SMCU, MCU, and DMCU.

	Area (microns)	%AR	Clock Cycle (nanoseconds)			%PP	Power Dissipation (watts)	%PDP
			PHI1	PHI2	Total			
SMCU	2788 x 2194	0	100	200	300	0	0.15	0
MCU	4480 x 2980	118	100	200	300	0	0.24	60
DMCU	4890 x 2980	138	100	250	350	17	0.25	67

%AR = Area Redundancy (extra area / the area of the SMCU)

%PP = Performance Penalty (increase in clock cycle / the clock cycle of the SMCU)

%PDP = Power Dissipation Penalty (increase in power dissipation / the power dissipation of the SMCU)

The MCU pays no performance penalty for CED. Error detection can be done with no interference in the normal operation. On the other hand, the DMCU has a performance penalty of 17%. The penalty is caused by the fact that check bits must be generated after address is available.

From the standpoint of area redundancy and performance penalty, the MCU is a slightly better design than the DMCU. The MCU has less area redundancy than the DMCU and has no performance penalty comparing to the SMCU. However, if the slight improvements in area redundancy and performance are not crucial to the chip requirements, the DMCU would be a better choice in term of the design and layout turn-around time. The turn-around time of the DMCU will be shorter than that of the MCU because there are no special layout constraints for designing the SMCU cell. Special layout constraints, as mentioned in Section 5.3, are effective only when placing the input and output checker after duplicating the SMCU cell.

CHAPTER 7

CONCLUSIONS

The microprogram control unit design proposed in this thesis provides a valuable method for on-chip concurrent error detection. The CED MCU requires more than a double the amount of chip area than that for a simplex MCU, but it does not have performance degradation. For CED, the MCU is a more favorable design than a duplicated MCU because the MCU has smaller area redundancy and better timing performance; however, under general conditions, the DMCU is a better choice because it offers better fault coverage, and is easier to design and to layout.

We plan to fabricate this layout. Once the chip is available, the design can go through hardware evaluation to check for the performance of the design.

There are many improvements that can be made on the MCU design especially in terms of the area redundancy. The duplication of the incrementer and the decrementer requires 13% and 23.9% extra areas, respectively. These numbers can be reduced by using totally self-checking incrementer and decrementer. Area redundancy can also be improved by including a second metal layer and by using careful layout techniques to minimize the amount of wasted areas.

Possible future research concerns inclusion of the retry capability in the chip so that transient errors can be automatically tolerated. Our design of an MCU would have less area redundancy because the duplicated control unit must be an MCU with its own retry capability and not an SMCU, for the DMCU to provide concurrent error detection. Another possibility for future research is the addition of ROM to the MCU to create a single chip total microprogram controller. The MCU approach may be more favorable than the DMCU approach because the area constraint is very important in this case.

APPENDIX A

BASIC CELLS

In the following few pages, basic cells for:

[1] Noninverting and inverting supper buffers.

[2] 4-input totally self-checking checker.

[3] Adders and subtractors.

[4] Register/Counter.

[5] Microprogram counter.

[6] Stack.

are shown in mixed notation or in block diagram.

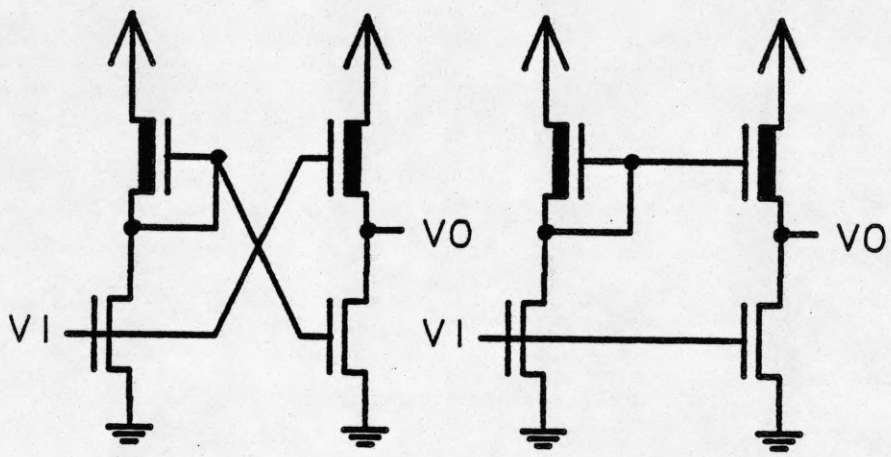

Figure A-1. Noninverting and Inverting Super Buffers (SBNI and SBI).

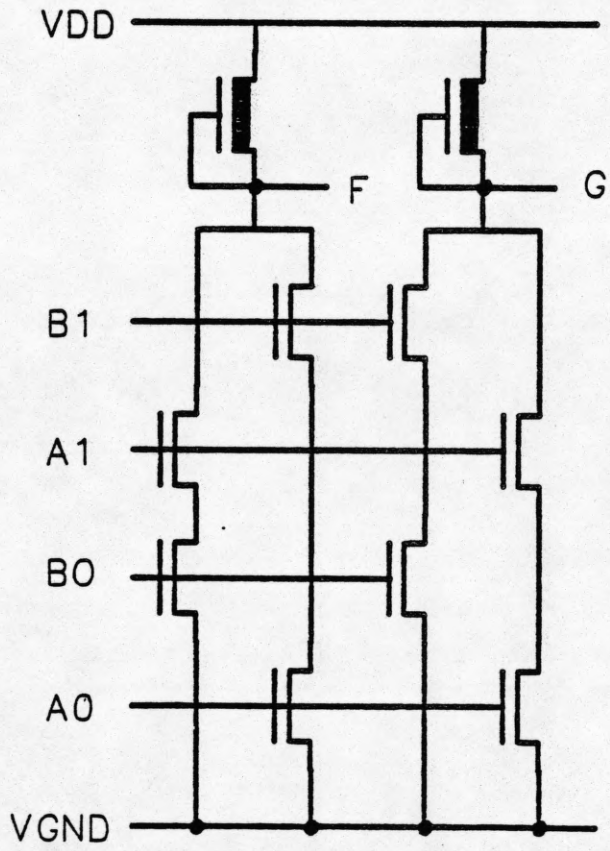

Figure A-2. 4-Input Totally Self-Checking Checker Cell.

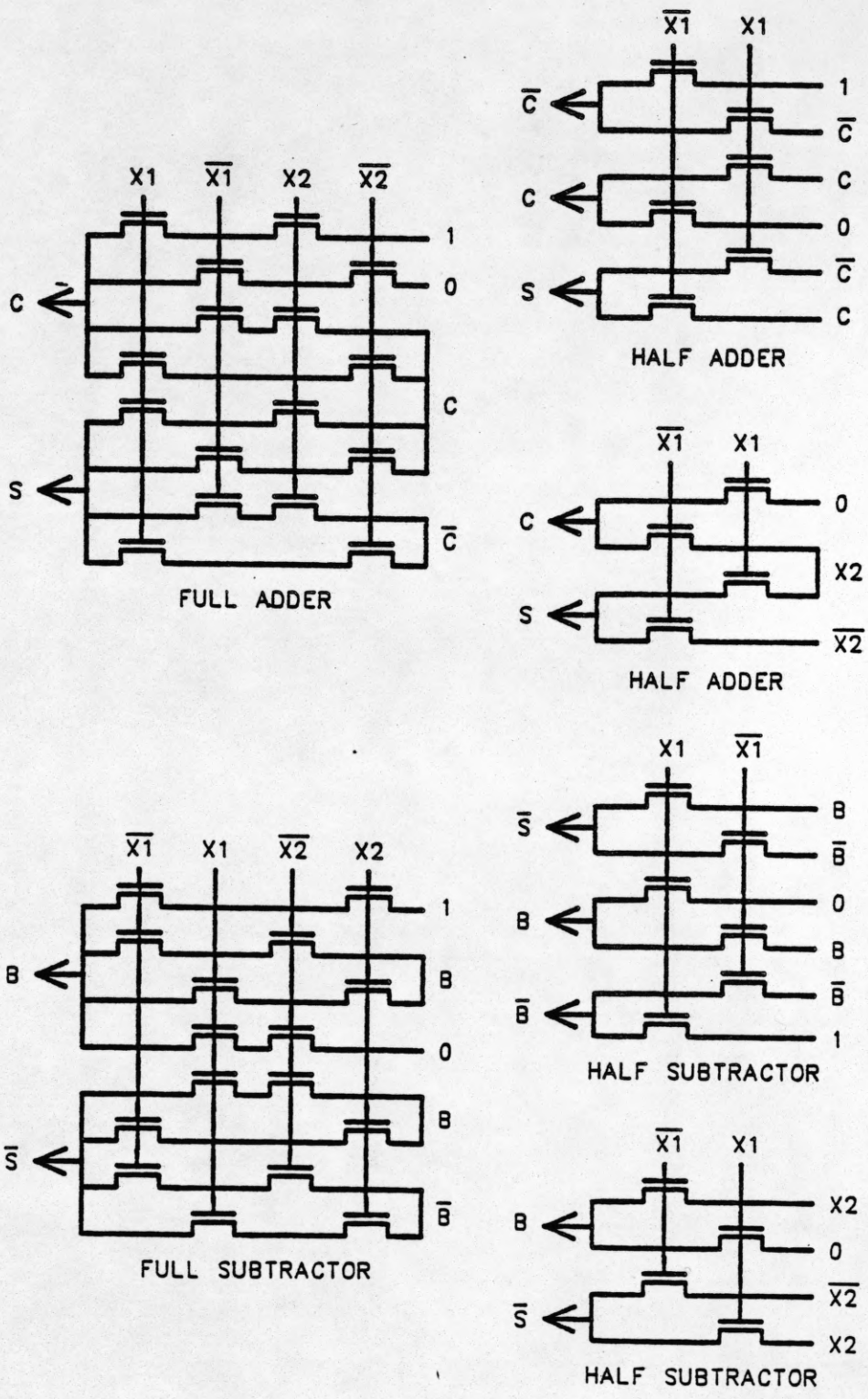

Figure A-3. Adder and Subtractor Cells.

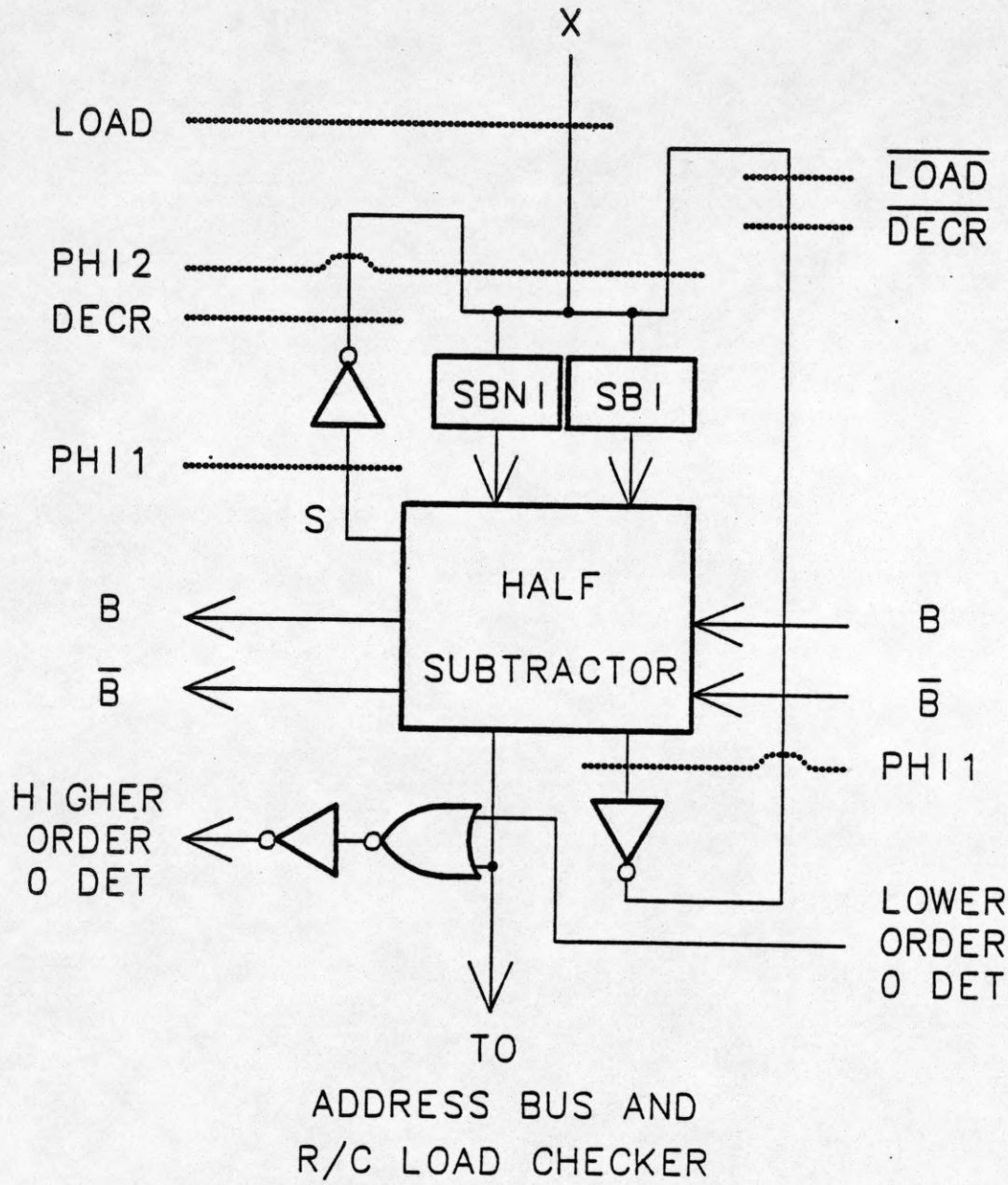

Figure A-4. Register/Counter Cell (RCCELL).

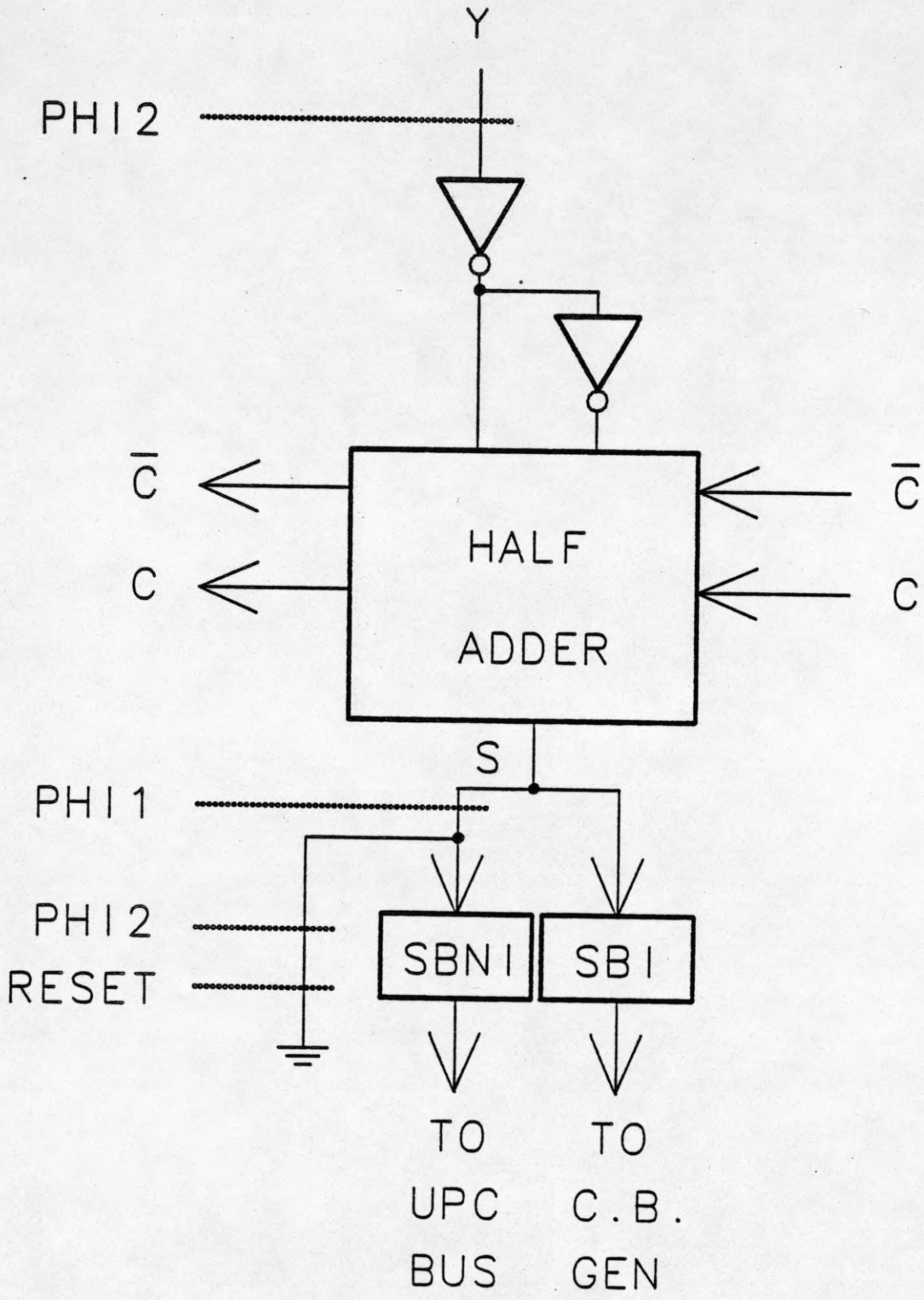

Figure A-5. Microprogram Counter Cell (UPCCELL).

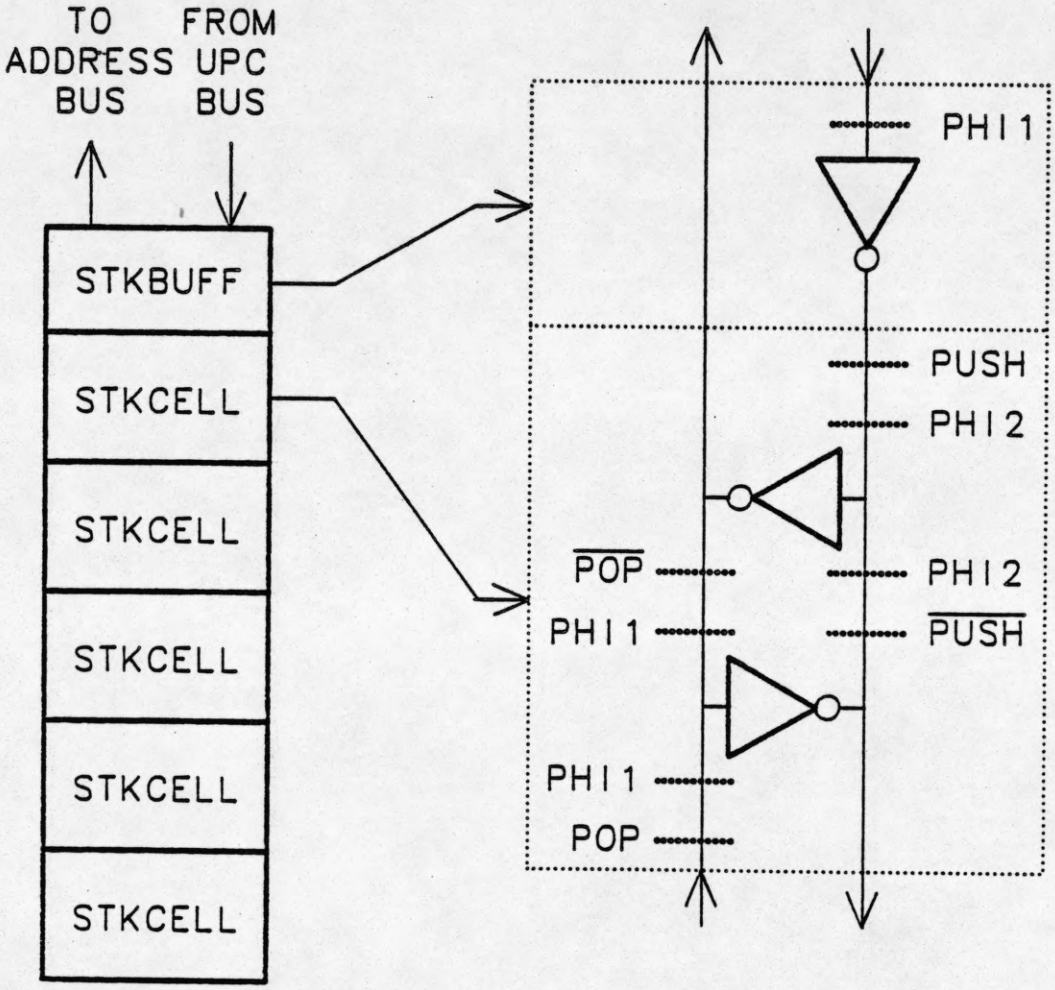

Figure A-6. Stack Cells.

APPENDIX B

INPUT AND OUTPUT PAD ASSIGNMENTS

The are a total of 52 input/output pads, and the pad assignments are shown in Table B-1. Each pad is assignmented with a number start in a clockwise motion from the bottom left corner to the bottom right of the chip, as shown in Figure 6-1.

Table B-1. Input/Output Pad Assignmants.

Signal	I/O	Pad Number	Comment
VGND	Input	1	
VDD	Input	14	
PHI1	Input	3	Clock phases
PHI2	Input	2	
$\overline{CC}$	Input	8	Condition code
CC	Input	7	
I3	Input	12	Instruction code
I2	Input	11	
I1	Input	10	
I0	Input	9	Least significant bit
ICB2	Input	30	Instruction code check bits
ICB1	Input	31	
ICB0	Input	32	Least significant bit
X11	Input	13	External address
X10	Input	15	
X9	Input	16	
X8	Input	17	
X7	Input	18	
X6	Input	19	
X5	Input	20	
X4	Input	21	
X3	Input	22	
X2	Input	23	
X1	Input	24	
X0	Input	25	Least significant bit
XCB3	Input	26	External input check bits
XCB2	Input	27	
XCB1	Input	28	
XCB0	Input	29	Least significant bit

Signal	I/O	Pad Number	Comment
VECT	Output	4	Enable signals
MAP	Output	5	
PL	Output	8	
ERROR1	Output	33	Dual-rail error signals
ERROR0	Output	34	from the PLA control checker
PHI1	Output	35	Clock phases
PHI2	Output	36	
Y11	Output	41	Address for the control-store
Y10	Output	42	
Y9	Output	43	
Y8	Output	44	
Y7	Output	45	
Y6	Output	46	
Y5	Output	47	
Y4	Output	48	
Y3	Output	49	
Y2	Output	50	
Y1	Output	51	
Y0	Output	52	Least significant bit
YCB3	Output	40	Address check bits
YCB2	Output	39	
YCB1	Output	38	
YCB0	Output	37	Least significant bit

REFERENCES

[Ande71] D. A. Anderson. "Design of Self-Checking Digital Networks Using Coding Techniques," R-527 Technical Report, Coordinated Science Laboratory, Urbana, Illinois, 1971.

[BaAb82] P. Banerjee and J. A. Abraham. "Fault Characterization of MOS VLSI Circuits," *Proceedings 1982 International Conference on Circuits and Computers*, New York, Sept. 29 - Oct. 1, 1982, pp.564-568.

[CSST73] R. W. Cook, W. H. Sisson, T. F. Storey and W. N. Toy, "Design of a Self-checking Microprogram Control," *IEEE Transactions on Computers*, vol. C-22, March 1973, pp.255-262.

[CrLa80] Y. Crouzet and C. Landrault, "Design of Self-Checking MOS-LSI Circuits: Application to a Four-Bit Microprocessor," *IEEE Transactions on Computers*, vol. C-29, no. 6, June 1980, pp. 532-537.

[DiSo75] M. Diaz and J. M. de Souza, "Design of Self-Checking Microprogram Controls," *Digest of International Symposium Fault-Tolerant Computing*, June 1975, pp.137-142.

[DuMa83] J. Duran and T. Mangir, "A Design Approach for a Microprogrammed Control Unit with Built in Self Test," *Proceedings of the 16th Annual Workshop on Microprogramming*, Sept. 1983, pp. 55-60.

[FuAb84] W. K. Fuchs and J. A. Abraham, "A Unified Approach to Concurrent Error Detection in Highly Structured Logic Arrays," *Proceedings of the 14th Annual International Symposium on Fault-Tolerant Computing*, Orlando, Florida, June 1984, pp. 4-9.

[HoSe80] R. W. Hon and C. H. Sequin, "A Guide to LSI Implementation," Technical Report SSL-79-7, XEROX Research Center, Palo Alto, California, 1980.

[IyKi82] S. V. Iyengar and L. L. Kinney. "Concurrent Testing of Flow of Control in Simple Microprogrammed Control Units," *Digest of the 1982 International Test Conference*, Cherry Hill, Nov. 1982, pp.469-479.

[JhAb84] N. Jha and J. A. Abraham, "MOS Implementation of Totally Self-Checking Circuits," To appear in: *Proceedings of the International Conference on Computer Design*, Oct. 1984.

[MaAD82] G. P. Mak, J. A. Abraham, and E. S. Davidson, "The Design of PLAs with Concurrent Error Detection," *Proceedings of the 12th International Symposium on Fault-Tolerant Computing*, Santa Monica, CA, June 1982, pp.303-310.

[Maki78] G. K. Maki. "A Self-Checking Microprocessor Design," *Journal of Design Automation and Fault-Tolerant Computing*, vol. 2, Jan. 1978, pp.15-27.

[MeCo80] C. Mead and L. Conway, *Introduction to VLSI Systems*, Reading: Addison-Wesley, 1980.

[MiBr80] J. Mick and J. Brick. *Bit-Slice Microprocessor Design*, New York: McGraw-Hill, 1980.

[Namj82] M. Namjoo, "Design of Concurrently Testable Microprogrammed Control Units," *Proceedings of the 15th Annual Workship on Microprogramming*, Palo Alto, CA, Oct. 1982, pp.173-180.

[PaFu82] J. H. Patel and L. Y. Fung, "Concurrent Error Detection in ALUs by Recomputing with Shifted Operands," *IEEE Transactions on Computers*, vol. C-31, July 1982, pp.589-595.

[SeLi80] R. M. Sedmak and H. L. Liebergot, "Fault Tolerance of a General Purpose Computer Implemented by Very Large Scale Integration," *IEEE Transactions on Computers*, vol. C-29, no. 6, June 1980, pp. 492-500.

[SrTh82] T. Sridhar and S. M. Thatte, "Concurrent Checking of Program Flow in VLSI Processors," *Digest of the 1982 International Test Conferene*, Cherry Hill, Nov. 1982, pp.191-199.

[TWMTS82] M. Tsao, A. Wilson, R. McGarity, C. Tseng, and D. Siewiorek, "The Design of C.fast: A Single Chip Fault Tolerant Microprocessor," *Digest of the International Symposium on Fault-Tolerant Computing*, June 1982, pp. 63-69.

[Wake78] J. Wakerly, *Error-Detecting Codes, Self-Checking Circuits, and Applications*, New York: North-Holland, 1978.

[WFAD83] C. Y. Wong, W. K. Fuchs, J. A. Abraham, and E. S. Davidson, "The Design of a Microprogram Control Unit with Concurrent Error Detection," *Proceedings of the 13th Annual International Symposium on Fault-Tolerant Computing*, Milan, Italy, June 1983, pp.476-483.

[Whit83] S. Whitaker, "Pass-Transistor Networks Optimize n-MOS Logic," *Electronics*, Sept. 22, 1983, pp.144-148.

[Will77] I. Williamson, "Design of Self-Checking and Fault-Tolerant Microprogrammed Controllers," *The Radio and Electronic Engineer*, vol. 47, Oct. 1977, pp.449-457.

[WiWi77] J. S. William and J. K. William. "A Compact High-Speed Parallel Multiplication Scheme," *IEEE Transactions on Computers*, vol. C-26, no. 10, Oct. 1977, pp.948-957.

[Wong82] C. Y. Wong, "The Design of a Microprogram Control Unit with Concurrent Error Detection," CSG-12 Technical Report, Coordinated Science Laboratory, Urbana, Illinois, 1982.