UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

RGB interface emulation with Infineon Aurix MCU

Relatore: Prof. / Dott Nicola Trivellin

Laureando: Ludovico Bonan

ANNO ACCADEMICO 2022 – 2023

Data di laurea 23/10/2023

To my parents
and friends

Abstract

This research investigates the emulation of the Digital Parallel RGB Interface using the Infineon Aurix™ TC3xx microcontroller, specifically the TC375 variant. Despite the advanced features of the microcontroller, there is an inherent lack of native support for this interface. This study attempts to address this limitation by using the General Timer Module (GTM) inherent to the Aurix™ family, with a primary focus on automotive applications.

An overview of Infineon and its MC department in Padua provides the basis and explains the motivation behind the research. A deeper understanding of the technological basis is established, highlighting the Aurix™ TC37x System on Chip Architecture, the GTM and the Digital Parallel RGB Interface. Key hardware and tools such as the Aurix™ TC375 Lite Kit Development Board, the Saleae 16 Pro Logic Analyzer, and the Tektronix 6 Series Oscilloscope are highlighted to illustrate their central role in the research process.

The research methodology is detailed and includes system design considerations and firmware implementation techniques. The primary objective is to emulate the RGB interface at a refresh rate of 100 Hz for 8-bit grayscale images. The results confirm the feasibility of the emulation effort. Functional tests verify correct system timing and signal integrity. However, certain constraints are revealed, particularly hardware limitations such as the challenges of transferring data to the GTM due to the System Peripheral Bus and the lack of flexibility in selecting the RGB interface clock.

Despite these challenges, the results confirm the potential of emulation using the Aurix™ TC375 microcontroller. The study concludes by highlighting the implications and offering avenues for future research, suggesting transitions to the TC39x series and the exploration of full-color RGB image support. The summary of this research highlights its relevance to advanced microcontroller-based interface emulation, particularly in the automotive domain.

Sommario

Questa ricerca studia l'emulazione dell'interfaccia RGB digitale parallela utilizzando il microcontrollore Aurix™ TC3xx di Infineon, in particolare la variante TC375. Nonostante le caratteristiche avanzate del microcontrollore, manca un supporto nativo per questa interfaccia. Questo studio cerca di risolvere questa limitazione utilizzando il General Timer Module (GTM) della famiglia Aurix-texttrademark, con particolare attenzione alle applicazioni automotive.

Una panoramica di Infineon e del suo dipartimento MC di Padova fornisce le basi e spiega le motivazioni alla base della ricerca. Viene approfondita la conoscenza delle basi tecnologiche, evidenziando l'architettura System on Chip Aurix™ TC37x, il GTM e l'interfaccia RGB digitale parallela. Hardware e strumenti chiave come la scheda di sviluppo TC375 Lite Kit di Aurix, l'analizzatore logico Saleae 16 Pro e l'oscilloscopio Tektronix serie 6 sono messi in evidenza per illustrare il loro ruolo centrale nel processo di ricerca.

La metodologia di ricerca è dettagliata e comprende considerazioni sulla progettazione del sistema e tecniche di implementazione del firmware. L'obiettivo primario è emulare l'interfaccia RGB a una frequenza di aggiornamento di 100 Hz per immagini in scala di grigi a 8 bit. I risultati confermano la fattibilità dello sforzo di emulazione. I test funzionali verificano la corretta temporizzazione del sistema e l'integrità del segnale. Tuttavia, sono emersi alcuni vincoli, in particolare le limitazioni hardware, come le difficoltà di trasferimento dei dati al GTM a causa del System Peripheral Bus e la mancanza di flessibilità nella selezione del clock dell'interfaccia RGB.

Nonostante queste difficoltà, i risultati confermano il potenziale dell'emulazione con il microcontrollore Aurix™ TC375. Lo studio si conclude evidenziando le implicazioni e proponendo strade per la ricerca futura, suggerendo il passaggio alla serie TC39x e l'esplorazione del supporto di immagini RGB a colori. La sintesi di questa ricerca evidenzia la sua rilevanza per l'emulazione avanzata di interfacce basate su microcontrollori, in particolare nel settore automobilistico.

Contents

List of Figures

List of Acronyms

ADC Analog to Digital Converte

AEI AE-Interface

ARU Advanced Routing Unit

ASIL Automotive Safety Integrity Level

ATOM ARU connected Timer Output Module

CCM Cluster Configuration Module

CMU Clock Management Unit

DAC Digital to Analog Converter

DSP Digital Signal Processor

DTM Dead Time Module

F2A FIFO to ARU

GTM General Timer Module

MC Microcontroller

MCAL Microcontroller Abstraction Layer

MCS Multi Channel Sequencer

MCU Microcontroller

PSM Paremeter Storage Module

RGB Red Green Blue

RTOS Real-Time Operating Systems

SPB System Peripheral Bus

TGPS TIM Gated Periodic Sampling Mode

TIM Timer Input Module

TOM Timer Output Module

1 Introduction

1.1 Context and motivation

The endeavor to advance automotive technology necessitates a continuous examination of existing microcontroller capabilities against the evolving requisites of the industry. I began exploring this area through three years of experience in Formula Student[1] team of the University of Padua and decided to continue with a six-month collaborative engagement with Infineon Technologies, specifically within the Microcontroller Department of the Automotive Division, to execute my master's thesis. This collaboration furnished a conducive platform for both academic and professional growth and provided a pivotal opportunity to address a critical industry need by leveraging Infineon's advanced technology.

Central to this endeavor was the objective to explore the feasibility of utilizing the General Timer Module (GTM), a Bosch IP integrated within the Aurix™ family of microcontrollers, to emulate the RGB interfacean functionality not natively supported by the device. The Aurix™ TC3xx microcontrollers are renowned for their robust and versatile features, making them a cornerstone in the automotive microcontroller landscape. However, the lack of a native RGB interface presented a limitation that necessitated an innovative workaround.

The GTM, originally designed for combustion engine control, emerged as a promising candidate for this emulation due to its inherent flexibility and con-

[1] Formula Student (often abbreviated as FS) is an international design competition for university students, where they design, build, test, and race small-scale formula-style racing cars.

figurability. Traditionally, the GTM has been employed in tasks necessitating precise timing and control, characteristics that are quintessential for effective engine management. However, its potential for applications beyond engine control remains largely unexplored, particularly in emulating the RGB interface which is pivotal for a myriad of automotive applications including, but not limited to, sophisticated lighting systems, display panels, and diagnostic interfaces.

The motivation to traverse this uncharted territory was propelled by a blend of academic inquisitiveness and the pragmatic aspiration to augment the functionality of Aurix™ microcontrollers. Furthermore, this endeavor exemplifies the numerous possibilities embedded within existing technologies, awaiting exploration and application. By venturing into this novel application of the GTM, this project not only aims at bridging a crucial functional gap but also embarks on a pathway that could potentially usher in a new wave of innovative solutions in automotive microcontroller technology.

The absence of any preceding attempts to utilize the GTM for RGB interface emulation underscored the novelty and the challenges inherent to this project. However, it was the essence of innovation and the prospect of contributing significant value to both the academic and automotive realms that propelled this project forward. Through experimentation and collaboration with Infineon, this thesis endeavors to unravel a new dimension of the GTM, thus paving the avenue for enhanced functionalities in automotive microcontrollers, and broadening the spectrum of what is achievable with existing technologies.

1.2 Infineon and the MC Department in Padua

Infineon is a leading semiconductor company that specializes in creating innovative microcontrollers for a wide range of applications. The company's microcontroller department has a long history of producing high-quality products that are known for their reliability, performance, and low power consumption. Infineon's microcontrollers are used in a variety of industries, including automotive, industrial, and consumer electronics [5].

One of the key strengths of Infineon's microcontroller department is its ability to offer a diverse range of products to meet the needs of different customers. The department's microcontrollers are based on a range of different architectures, including ARM Cortex-M, TriCore, and XMC. This allows the company to offer a wide range of products that can be customized to suit individual requirements.

Infineon is a major supplier of microcontrollers to the automotive industry, and its products are used by many of the world's leading automotive manufacturers. The company's microcontrollers are used in a variety of automotive applications, including engine management, transmission control, and safety systems.

In addition to automotive applications, Infineon's microcontrollers are used in a range of industrial and consumer electronics applications. These include factory automation, robotics, home automation, and IoT devices. Infineon's microcontroller department has a strong focus on innovation and research, and it is constantly developing new products and technologies to meet the changing needs of the market.

Infineon is also committed to sustainability and has developed a range of microcontrollers that are designed to be energy-efficient and environmentally friendly. The company has a global presence, with offices and manufacturing facilities located in Europe, Asia, and the Americas.

In particular, the Microcontroller Department in Padua specializes in Technical Marketing, advanced technical support to Infineon clients, and embedded Flash testing

1.3 Thesis objectives

The primary objective of this thesis is to evaluate the potential for leveraging the Aurix™ family of microcontrollers to assess the feasibility of emulating an RGB interface. Should this emulation prove feasible, the intention is to develop a prototype capable of transmitting 8-bit grayscale images. The image data structure under analysis consists of 64 rows, each comprising 256 pixels, with a targeted refresh rate of 100 Hz.

Equipped with a comprehensive array of features and functionalities, the Aurix™ microcontrollers serve as an optimal platform for the investigation of advanced methods aimed at overcoming constraints associated with the emulation of an RGB interface. Central to this investigation is the General Timer Module (GTM), an intellectual property of Bosch, which exhibits significant potential for facilitating such emulation due to its inherent flexibility and configurable parameters.

1.4 Document structure

The master thesis is methodically organized into a sequence of seven major chapters that detail specific facets of the conducted research.

Chapter 1, titled "Introduction", establishes the broader research context. It begins by introducing the motivation behind the research, followed by an exposition about Infineon and its MC Department in Padua. This chapter culminates by outlining the thesis objectives.

In Chapter 2, "Theoretical and Technological Foundations", there's a deep dive into the technical foundation underpinning the study. It gives prominence to the Infineon Aurix™ microcontroller, discussing microcontrollers in general, the Aurix™ microcontroller family in particular, and an in-depth review of the Aurix™ TC37x System on Chip Architecture. The intricacies of the Generic Timer Module (GTM) and its components are explored in detail, alongside a section dedicated to the Digital Parallel RGB Interface.

Chapter 3 is designated "Hardware and Tools". Here, the hardware and tools pivotal to the study are discussed. The Aurix™ TC375 Lite Kit Development Board is examined extensively, with emphasis on its associated toolchain and debugger. Instruments like the Saleae 16 Pro Logic Analyzer with its Logic 2 Software and the Tektronix 6 Series Oscilloscope are also elaborated upon.

The subsequent, Chapter 4, "Methodology", elucidates the methodological approach employed in the research. This encompasses an initial requirements analysis, a delve into the system design, and a rigorous exploration of firmware implementation techniques, including the GTM configuration and CPU preprocessing.

Chapter 5, "Testing and Validation", articulates the validation procedures and methodologies of the research. This entails a preliminary evaluation employing the Saleae Logic Analyzer, a more profound analysis utilizing the Tektronix 6 Series Oscilloscope, and the ultimate validation steps conducted with the Characterization Board.

"Results and Discussion" is the focus of Chapter 6. This chapter presents the findings of the research and provides an interpretive discussion. It delves into the detailed analysis of the results, a segment on encountered limitations, and subsequently branches into practical applications, spotlighting areas such as Advanced Front-Light Systems.

The thesis wraps up with Chapter 7, "Conclusions and Future Works". This concluding chapter summarizes the research's pivotal findings, draws definitive conclusions based on the gathered data and analysis, and offers insights into possible future works and the potential expansion of the research.

2

Theoretical and Technological Foundations

2.1 Infineon Aurix™ microcontroller

2.1.1 Overview of microcontrollers

Microcontrollers are highly integrated computing devices encapsulated within a single microchip, serving as the linchpin of contemporary embedded systems. The architecture of a microcontroller typically comprises a processor, memory, and input/output (I/O) peripherals, adept at executing instructions housed within their memory to administer tasks and manipulate data.

These devices are engineered for versatility and adaptability, encompassing key features that resonate across a broad spectrum of electronic applications. Their compact design facilitates the integration of all crucial components onto a single chip, thereby minimizing the physical space requisite for computing functionalities. Operating under low power conditions, microcontrollers are ideal for battery-powered and energy-sensitive applications. They are often deployed in real-time applications, where timely responses and performance are critical. Additionally, microcontrollers are equipped with embedded peripherals such as timers, communication ports, and Analog-to-Digital Converters (ADCs)/Digital-to-Analog Converters (DACs), which further bolster their functionality. The programmable nature of microcontrollers enables customization, allowing users to tailor operations to specific requirements using high-level languages like C, and Python, or resorting to assembly language for finer control.

The efficacy of a microcontroller is intrinsically tethered to its internal architecture. Central Processing Units (CPUs) like ARM, AVR, or PIC constitute the brain of the microcontroller, executing program instructions. Memory is bifurcated into Flash for program code storage, SRAM for data storage, and EEPROM for non-volatile data storage, each playing a pivotal role. I/O ports facilitate interactions with external devices or components, while auxiliary features like timers, counters, communication interfaces, and ADCs/DACs augment the microcontrollers capabilities.

Microcontrollers are the quintessence of electronics and embedded systems, serving as the nucleus of specialized computer systems designated for specific tasks. They are integral to a myriad of machines we engage with daily, from humble home appliances to complex systems like medical apparatus and automobiles. The burgeoning realm of the Internet of Things (IoT) has accentuated the significance of microcontrollers. IoT devices, necessitating real-time processing, compactness, and energy efficiency, are predominantly built around microcontrollers. This underscores the indispensable role microcontrollers play not only in our daily electronics but also at the vanguard of technological evolution.

2.1.2 Aurix™ microcontroller family

Infineon Aurix™ is a family of high-performance microcontrollers (MCUs) designed and manufactured by Infineon Technologies, primarily for automotive applications and other sectors such as industrial automation, motor control, power systems, and functional safety. From now on, we will refer specifically to the Aurix™ TC3xx line, which represents the second generation of the family.

The Aurix™ family is based on a scalable multicore architecture with up to six independent cores, achieving up to 690 DMIPS (Dhrystone Million Instructions per Second) per core. These cores are built on the Infineon TriCore architecture, which balances processing performance, energy efficiency, and production costs by combining 32-bit RISC (Reduced Instruction Set Computer) architecture, Digital Signal Processor (DSP), and microcontroller elements.

Key aspects include functional safety support, compliance with international ISO 26262 standards for automotive electronic safety systems, and enabling designers to develop applications meeting the highest functional safety requirements (Automotive Safety Integrity Level - ASIL[1]) while ensuring reliability in critical missions and control systems [2].

Distinctive features include real-time processing support, essential for automotive and motor control applications. The architecture incorporates an advanced timing and performance control system, enabling accurate task and priority management while respecting timing constraints.

The Infineon Aurix™ series is a high-performance, functionally safe microcontroller solution designed to meet advanced application needs in the automotive sector and beyond. Its scalable multicore architecture, functional safety support, and extensive range of integrated peripherals and interfaces enable it to tackle various design and development challenges, such as real-time processing, complex system control, and power management.

A primary advantage is its flexibility, allowing designers to select the optimal core, memory, and peripheral configuration for their project's specific requirements. This flexibility increases design efficiency, reducing development time and associated costs.

In the context of technological advancements in the automotive sector, such as autonomous driving and vehicle electrification, the Aurix™ series is particularly suited to address the challenges posed by these new technologies. For example, it can implement advanced control algorithms for autonomous driving, manage complex electric traction systems, and monitor battery health and status in electric vehicles.

Furthermore, the Aurix™ series supports real-time operating systems (RTOS) and industry-specific development tools such as AUTOSAR MCAL[2], facilitating microcontroller integration into existing systems and accelerating development. Infineon also provides a rich support ecosystem, including software libraries, code examples, and documentation, to help designers fully leverage its capabilities.

2.1.3 Aurix™ TC37x System on Chip Architecture

During the requirements analysis, the Aurix™ TC 37x System-on-Chip (SoC) was identified as the optimal candidate for firmware implementation. This

[1] It is a risk classification system defined by the ISO 26262 standard for the functional safety of road vehicles.

[2] Microcontroller Abstraction Layer: a standardized software interface for automotive ECUs

choice was based on a rigorous evaluation of the SoC's capabilities in relation to the specified requirements, thereby ensuring its suitability for the intended application.

At its core lies a multicore structure composed of three 32-bit TriCore processing units, each capable of operating at speeds of 300 MHz. A dedicated hardware security module (HSM) is also integrated within the system to deliver robust cryptographic protection.

The TriCore processing units, which serve as the foundation of the Aurix™ system architecture, are built upon the first unified, single-core, 32-bit microcontroller-DSP (Digital Signal Processor) architecture, tailored explicitly for real-time embedded systems. The TriCore Instruction Set Architecture (ISA) merges the real-time capabilities of a microcontroller, the computational prowess of a DSP, and the high performance-to-price attributes of a RISC (Reduced Instruction Set Computer) load/store architecture [4] into a compact, reprogrammable core. Furthermore, the ISA supports a uniform 32-bit address space, incorporating optional virtual addressing and memory-mapped I/O capabilities.

This versatile architecture enables a broad spectrum of implementations, extending from scalar to superscalar configurations, and can effortlessly interact with various system architectures, including multiprocessing environments. This inherent flexibility at both the implementation and system levels paves the way for customizable trade-offs between performance and cost at any given moment. The TriCore ISA supports 16-bit and 32-bit instruction formats, with the 16-bit instructions functioning as a subset of the 32-bit instructions. These instructions can significantly reduce code space, leading to lower memory requirements and decreased system power consumption.

One of the primary factors determining real-time responsiveness, an essential aspect for automotive and industrial applications, is the interplay between interrupt latency and context-switch time. The high-performance TriCore architecture minimizes interrupt latency by circumventing lengthy multi-cycle instructions and incorporating a flexible hardware-supported interrupt scheme. Additionally, the architecture is designed to facilitate rapid context switching, further enhancing its real-time performance capabilities.

To ensure efficient and secure communication between cores and subsystems, Aurix™ features two internal bus: System Resource Interconnect Fabric (SRI Fabric) that connects the TriCore CPUs, the DMA module, and other high bandwidth requestors to high bandwidth memories and other resources for instruction fetches and data accesses and the System Peripheral Bus (SPB) enables parallel communication between cores, peripherals, and DMA module, reducing latency and improving overall system performance. In addition, the Aurix™ TC37x integrates a wide range of peripherals and interfaces to support a variety of functions, such as motor control, battery management, network communication, and sensor monitoring. Some of the main peripherals include CAN FD (Controller Area Network Flexible Data rate), Ethernet, LIN (Local Interconnect Network), FlexRay, UART (Universal Asynchronous Receiver/Transmitter), SPI (Serial Peripheral Interface), and I2C (Inter-Integrated Circuit) interfaces.

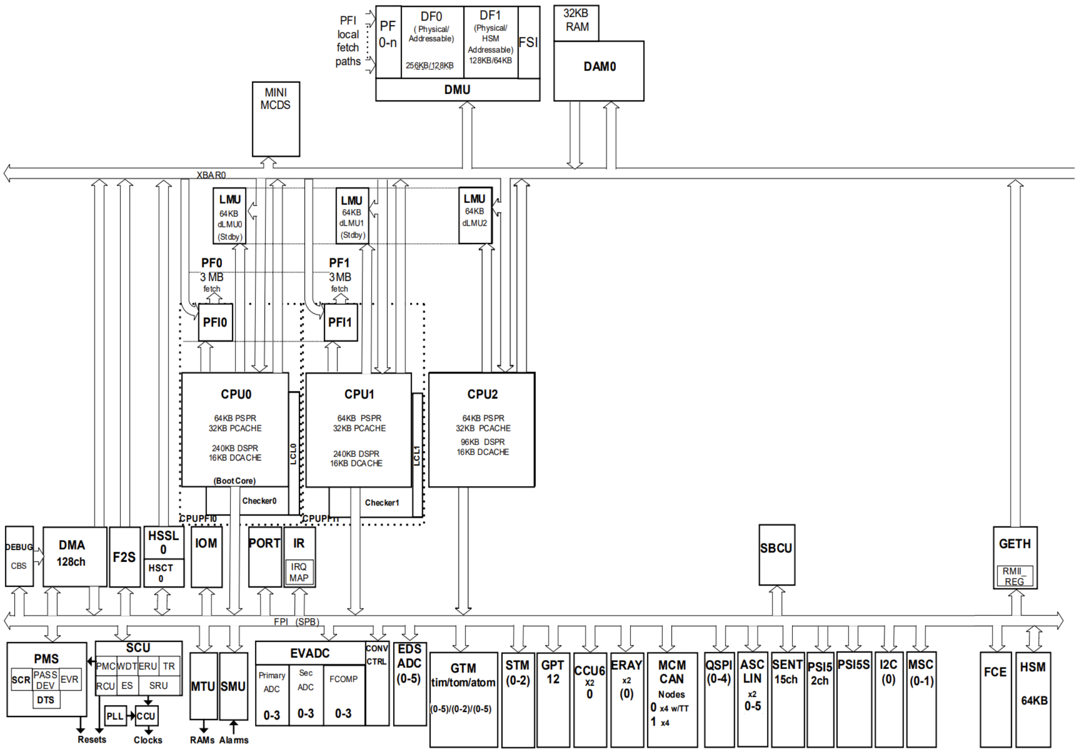


Figure 2.1: Aurix™ TC37x architecture [2]

In the context of this master's thesis, the Generic Timer Module (GTM) holds a pivotal role. Though integrated into Aurix™, this peripheral is an Intellectual Property developed by Bosch. It will be our primary component of interest, and we will provide a comprehensive exploration of its functionality in the upcoming section.

2.1.4 Generic Timer Module (GTM)

2.1.4.1 GTM Architecture

The Generic Timer Module (GTM) is a crucial component of the Aurix™ TC3xx system. It is designed to serve different application domains and different classes within these application domains. The GTM is comprised of two main parts: the GTM IP, designed by Bosch, and the GTM Wrapper, designed by Infineon. The GTM forms a generic timer platform that is highly scalable and configurable. This is achieved through an architecture philosophy where dedicated hardware sub-modules are located around a central routing unit, referred to as the Advanced Routing Unit (ARU). The ARU can connect the sub-modules in a flexible manner, with the connectivity being software programmable and configurable during runtime.

The GTM architecture is divided into clusters, each containing different sub-modules such as ATOM, MCS, PSM, TIM, TOM, and DTM. The number of sub-modules and the number of channels per sub-module can vary depending on the requirements of the application domains. The Parameter Storage Module (PSM) and the Cluster Dead Time Module (CDTM) are virtual hierarchies consisting of various sub-modules. The PSM consists of the sub-modules F2A, FIFO, and AFD, while the CDTM consists of up to six DTM modules. The availability of the DTM instances depends on the GTM device configuration. These sub-modules cater to a broad range of applications relevant to a timer module. Key features include Central/Edge aligned PWM Generation, asymmetric dead time generation, support for DC-DC phase shift operation, a logical combination of digital signals, and provision of a common time base for the system. It also enables multiple capture/compare of external signals, complex digital signals generation, digital signals filtering and characterization, and input signal measurement. The GTM IP supports BLDC motor control using Block Commutation Mode, engine angle clock for engine management applications, and injection/ignition pulses generation with dedicated hardware support [3]. It also includes programmable RISC-like cores to offload the CPU, automatic data sharing between GTM modules without CPU/DMA involvement, and a configurable operating frequency up to 200MHz. Here we will outline in more detail the sub-modules used to achieve the goal.

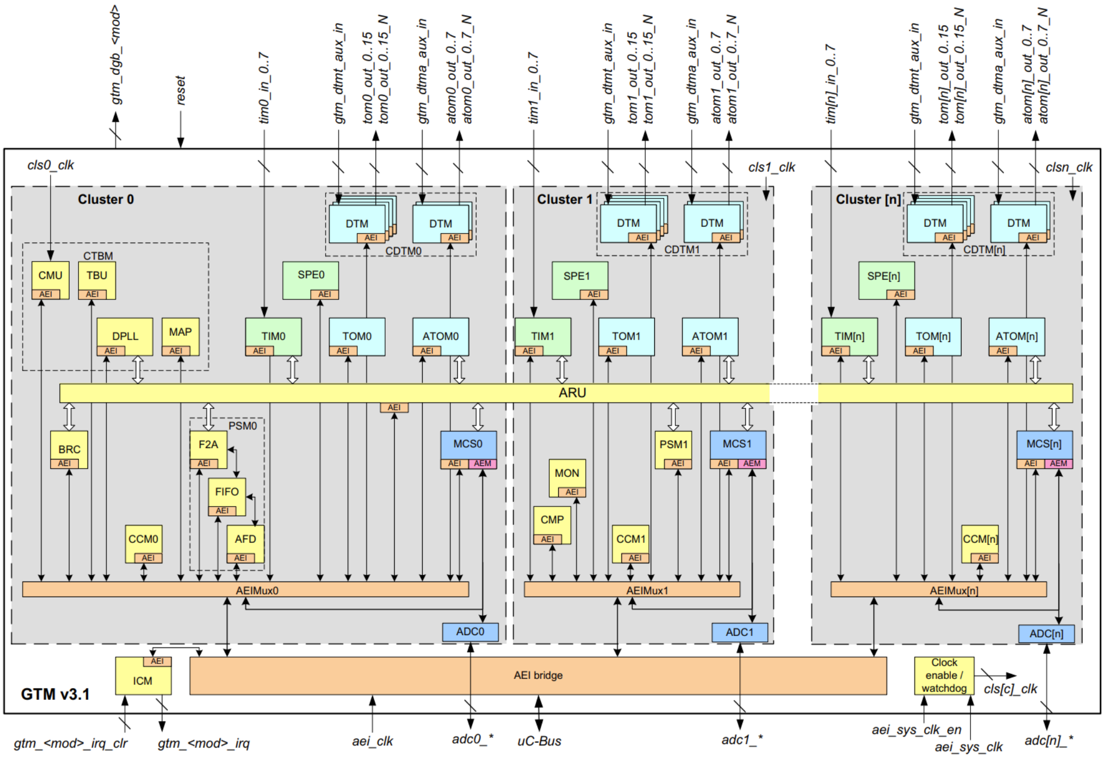


Figure 2.2: GTM architecture [3]

2.1.4.2 Clock Management Unit (CMU)

The Clock Management Unit (CMU) is a critical component within the Generic Timer Module (GTM) of the Aurix™ TC3xx. It is primarily responsible for generating the clock signals for the counters and the GTM. The CMU comprises three sub-units, each generating different clock sources utilized across the GTM. The primary clock source for this sub-module is the cluster 0 clock signal, also known as cls0_clk. This signal is defined by the value of the CLS0_-CLK_DIV bit field in the GTM_CLS_CLK_CFG register. This clock signal serves as the foundational time base for the operations within the GTM.

The Configurable Clock Generation Unit (CFGU) is one of the three sub-units within the CMU. It provides eight dedicated clock sources to various GTM modules, including TIM, ATOM, TBU, and MON. Each instance of these modules can select an arbitrary clock source, thereby allowing for a wide range of time bases to be specified. This flexibility is crucial for the diverse timing requirements of the different modules within the GTM.

The Fixed Clock Generation (FXU) sub-unit is another integral part of the CMU. It generates predefined, non-configurable clocks, denoted as CMU_FX-

CLK[y] (where y ranges from 0 to 4), for the TOM and MON modules. These clock signals are derived from the CMU_GCLK_EN signal generated by the Global Clock Divider, or from one of the eight CMU_CLK[x] enable signals, depending on FXCLK_SEL. The dividing factors for these signals are defined as 2^0, 2^4, 2^8, 2^{12}, and 2^{16}, providing a range of clock frequencies for different applications.

The External Clock Generation (EGU) sub-unit is capable of generating up to three external chip clock signals, visible at CMU_ECLK[z] (where z ranges from 0 to 2), with a duty cycle of approximately 50%. The EGU sub-unit can also generate a clock signal, CMU_CLK8, for the module CCM to manage two clock domains.

The clock source signals CMU_CLK[x] (where x ranges from 0 to 7) and CMU_FXCLK[y] are implemented in the form of enable signals for the corresponding registers. This means that the actual clock signal of all registers always uses the CLS0_CLK signal, ensuring synchronization across the system.

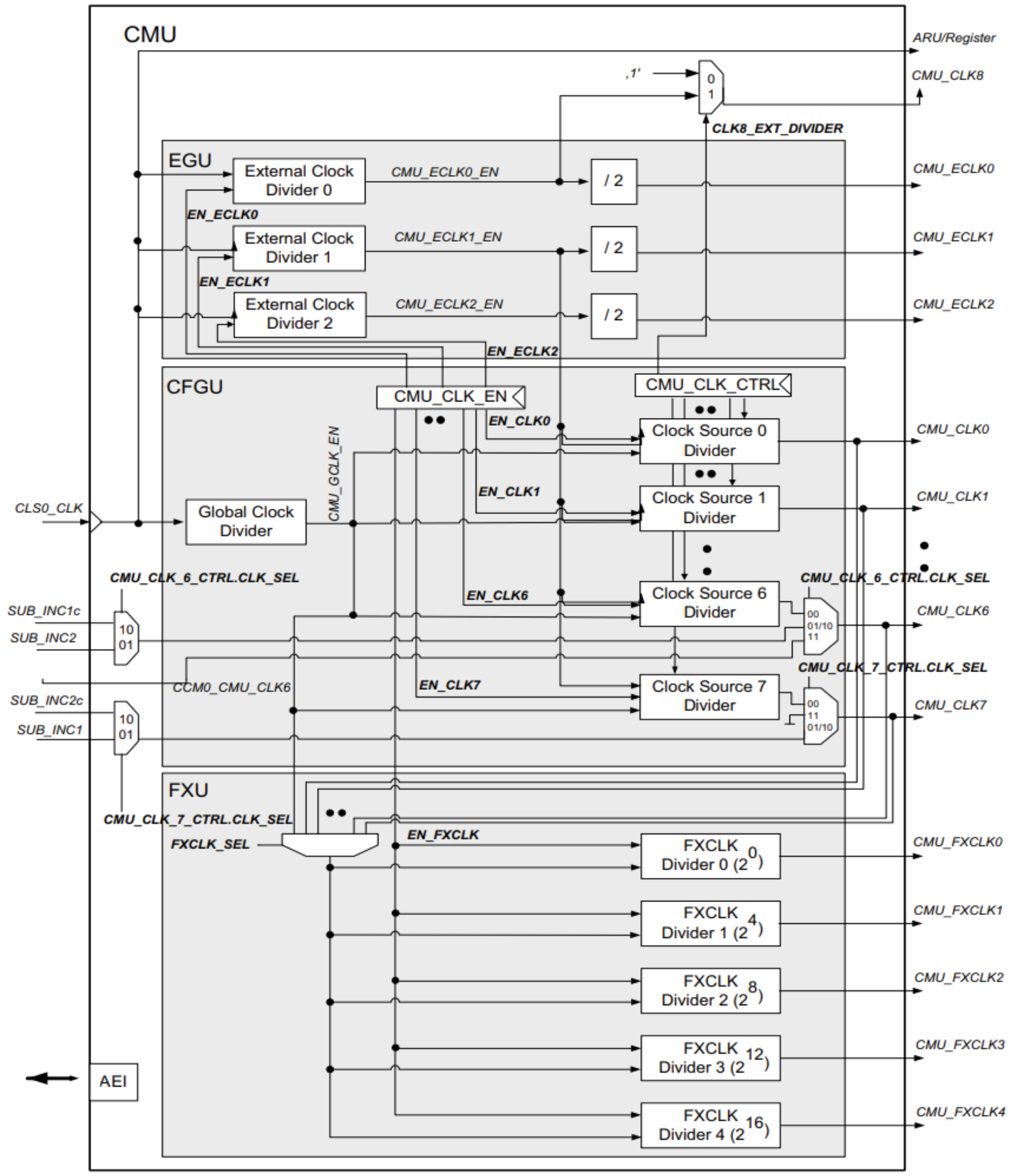


Figure 2.3: GTM CMU architecture [3]

2.1.4.3 Parameter Storage Module (PSM)

The Parameter Storage Module (PSM) is a virtual hierarchy comprising several sub-modules, including F2A, FIFO, and AFD. The FIFO (First In First Out) sub-module is the storage part of the PSM and interfaces with the ARU and AEI bus through the F2A (FIFO to ARU Unit) and the AFD (AEI to FIFO Data Interface). The FIFO RAM counts 1K words, with a word size of 29 bits. This gives the freedom to program or receive 24 bits of data with the five control bits inside an ARU data word. It houses eight logical FIFOs that are configurable in size, operating modes, and fill level control. These logical FIFOs facilitate a data stream between the GTM's sub-modules and the microcontroller connected to the AFD sub-module. The F2A sub-module serves as the interface between the ARU and the FIFO sub-module. Given the data width discrepancy between the ARU (53 bits) and the FIFO (29 bits), the F2A is designed to distribute the data to and from the FIFO channels in a configurable manner. The data transfer between FIFO and ARU is organized with eight different streams connected to the eight channels of the corresponding FIFO module. Each stream can be enabled or disabled separately in the F2A[i]_ENABLE register. The F2A unit provides several transfer modes to map the 29-bit data of the FIFO to/from the 53-bit data of the ARU. The AFD sub-module implements a data interface between the AEI bus and the FIFO sub-module. The AFD sub-module provides one buffer register dedicated to the logical channels of the FIFO, so access to the corresponding FIFO channel is given by reading or writing this buffer register AFD[i]_CH[x]_BUF_ACC. An AEI write access to the buffer register where the corresponding FIFO channel is full will be ignored, and the data will be lost. Conversely, an AEI read access to the buffer register where the corresponding FIFO channel is empty will be served with zero data.

2.1.4.4 Timer Input Module (TIM)

The Timer Input Module (TIM) is responsible for filtering and capturing input signals of the GTM. It measures several characteristics of the input signals inside its channels. These characteristics can be timestamp values of detected input rising or falling edges along with the new signal level, the number of edges received since the channel was enabled with the current timestamp or Pulse Width Modulation (PWM) signal duration for a whole PWM period. The detected input characteristics of the TIM module can be routed through the ARU to subsequent processing units of the GTM.

The TIM operates in several modes, each with its unique functionality, here are some of the main ones:

1. **TIM PWM Measurement Mode (TPWM):** In this mode, the TIM channel measures the duty cycle and period of an incoming PWM signal. The DSL bit defines the polarity of the PWM signal to be measured. The measurement starts either after the filter detects the first rising edge or immediately after activating the channel by TIM_EN=1, depending on the IMM_START setting.

2. **TIM Input Event Mode (TIEM):** The TIM channel counts edges in this mode. It is configurable if rising, falling, or both edges should be counted. This can be done with the bit fields DSL and ISL in the TIM[i]_CH[x]_CTRL register.

3. **TIM Input Prescaler Mode (TIPM):** In TIPM, the number of edges to be detected before a TIM[i]_NEWVAL[x]_IRQ (New Value Interrupt Request) is raised can be programmed. The edges to be counted can be selected by the bit fields DSL and ISL of register TIM[i]_CH[x]_CTRL.

4. **TIM Bit Compression Mode (TBCM):** This mode can be used to combine all filtered input signals of a TIM sub-module to a parallel m bit data word, which can be routed to the ARU, where m is the number of channels available in the TIM sub-module.

5. **TIM Gated Periodic Sampling Mode (TGPS):** The number of CMU clock cycles that should elapse before capturing and raising TIM[i]_NEWVAL[x]_IRQ is programmable. In this mode, it must be specified in the CNTS register after how many CMU clock cycles the interrupt has to be raised.

Each mode provides a unique way of processing and interpreting input signals, making the GTM highly versatile and adaptable to various applications.

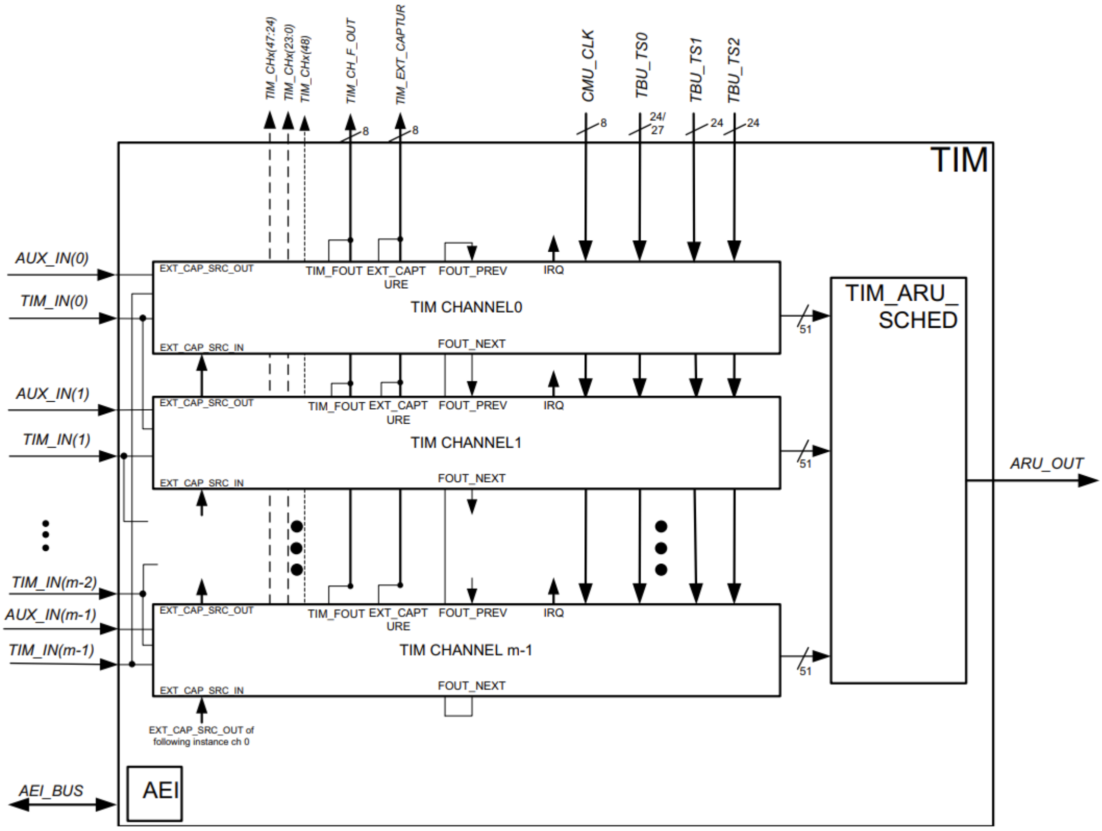


Figure 2.4: GTM TIM architecture [3]

2.1.4.5 Timer Output Module (TOM)

Another key module within the GTM is the Timer Output Module (TOM), designed to generate simple Pulse Width Modulation (PWM) signals. This chapter provides a comprehensive overview of the TOM, focusing on its operation and functionality. The TOM offers up to 16 independent channels to generate PWM signals at each output pin TOM[i]_CH[x]_OUT. Additionally, a pulse count modulated signal can be generated at the TOM output TOM[i]_CH15_OUT. The TOM channel count, denoted as cCTO, refers to the number of channels per instance minus one. The TOM module incorporates two global channel control units, TGC0 and TGC1. These units control the enabling and disabling of the channels and their outputs, as well as the update of their period and duty cycle register. The TOM receives two (three) timestamp values, TBU_TS0, TBU_TS1 (and TBU_TS2), to realize synchronized output behavior based on a common time base. The TOM sub-module supports four different kinds of signaling mechanisms: global enable/disable mechanism for each TOM channel, global output enable mechanism for each TOM channel, global force update mechanism for each TOM channel, and update enable of the register CM0, CM1, and CLK_SRC for each TOM channel. Each TOM channel is comprised of a Counter Compare Unit 0 (CCU0) that holds the counter register CN0 and the period register CM0, a Counter Compare Unit 1 (CCU1) that houses the duty cycle register CM1, and a Signal Output Generation Unit (SOU) that includes the output register SOUR. The output of the individual TOM channels can be controlled using the register TOM[i]_TGC[y]_OUTEN_CTRL and TOM[i]_TGC[y]_OUTEN_STAT. If a TOM channel is disabled by the register TOM[i]_TGC[y]_OUTEN_STAT, the actual value of the channel output at TOM_CH[x]_OUT is defined by the signal level bit (SL) defined in the channel control register TOM[i]_CH[x]_CTRL. If the output is enabled, the output at TOM_CH[x]_OUT depends on the value of flip-flop SOUR. The register TOM[i]_TGC[y]_FUPD_CTRL defines which of the TOM channels receive a FORCE_UPDATE event if the trigger signal CTRL_-TRIG is raised. The force update request is stored and executed synchronized to the selected CMU_FXCLK. The register bits UPEN_CTRL[z] define for which TOM channel the working register CM0, CM1, and CLK_SRC update by the corresponding shadow register SR0, SR1, and CLK_SRC_SR is enabled.

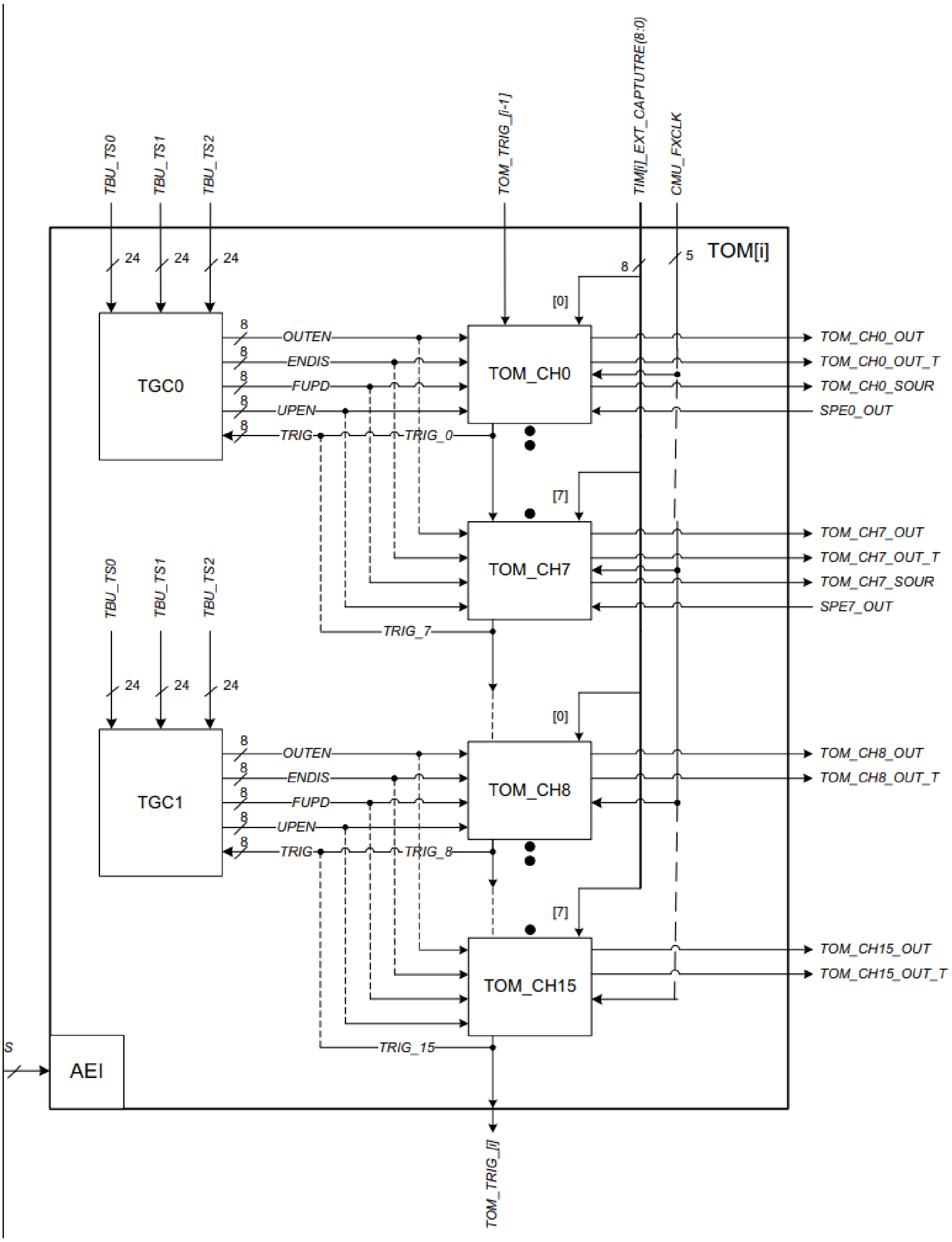


Figure 2.5: GTM TOM architecture [3]

2.1.4.6 ARU connected Timer Output Module (ATOM)

The ARU-connected Timer Output Module (ATOM) is designed to generate complex output signals without CPU interaction, thanks to its connectivity to the ARU. This functionality is typically facilitated by output signal characteristics provided over the ARU connection through other modules such as the MCS, DPLL, or PSM. Each ATOM module houses eight output channels, each capable of operating independently in several configurable operation modes. The ATOM's architecture is akin to the TOM module, with a few notable differences. The ATOM integrates only eight output channels and includes an ATOM Global Control sub-unit (AGC) for the ATOM channels. The ATOM is connected to the ARU and can set up individual read requests from the ARU and write requests to the ARU. The ATOM compare registers are 24-bit wide instead of only 16 bits. Moreover, the ATOM channels can generate signals based on time stamps and produce a serial output signal on behalf of an internal shift register. Another significant difference to TOM is that the input clocks for the ATOM channels originate from the configurable CMU_CLKx signals of the CMU module, providing the flexibility to select a programmable input clock for the ATOM channel counters. The ATOM channels provide five modes of operation, which are as follows

1. **ATOM Signal Output Mode Immediate (SOMI):** The ATOM channels generate an output signal immediately after receiving an ARU word according to the two signal-level output bits of the ARU word received through the ACBI bit field.

2. **ATOM Signal Output Mode Compare (SOMC):** The ATOM channel generates an output signal on behalf of time stamps located in the ATOM operation registers. These time stamps are compared with the time stamps the TBU generates.

3. **ATOM Signal Output Mode PWM (SOMP):** The ATOM channel can generate simple and complex PWM output signals like the TOM submodule by comparing its operation registers with a sub-module internal counter.

4. **ATOM Signal Output Mode Serial (SOMS):** The ATOM channel generates a serial output bit stream on behalf of a shift register. The number of

bits shifted, and the shift direction is configurable. The shifting frequency is determined by one of the CMU_CLKx clock signals.

5. **ATOM Signal Output Mode Buffered Compare (SOMB):** The ATOM channel generates an output signal on behalf of time stamps located in the ATOM operation registers. These time stamps are compared with the time stamps the TBU generates.

The ATOM channel also features an operation and shadow register set. This architecture allows for simultaneous operation with the operation register set, while the shadow register set can be reloaded with new parameters over CPU or ARU. The ATOM module supports four different signaling mechanisms, and the ATOM channels have an ARU Communication Interface (ACI) sub-unit responsible for data exchange from and to the ARU. This is done with the two implemented registers SR0, SR1, and the ACBI and ACBO bit fields that are part of the ATOM[i]_CH[x]_STAT register.

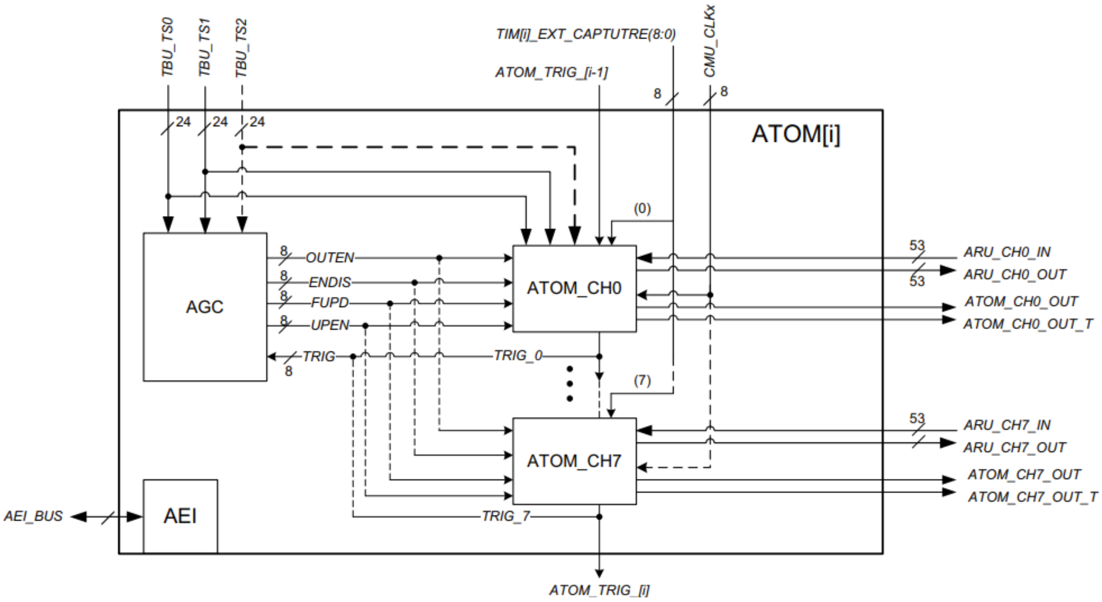


Figure 2.6: GTM ATOM architecture [3]

2.2 Digital Parallel RGB Interface

The digital parallel Red-Green-Blue (RGB) display interface is a fundamental digital imaging and display system technology. It operates by transmitting color information from a source, such as a computer's graphics card, to a target display, including monitors or television screens. The RGB interface finds its root in the additive color model, wherein varying intensities of red, green, and blue are combined to create a wide range of perceivable colors. This chapter aims to provide a comprehensive introduction to the digital parallel RGB interface, covering its fundamental concept, operation, and key components.

A digital parallel RGB interface communicates color information via separate data lines for each of the three color channels. This interface's "parallel" nature refers to its ability to transmit multiple bits of data simultaneously across these lines. This feature enables high-speed data transfer, a requisite for real-time, high-resolution video display. However, this requires wider data buses than serial or SPI interfaces, which can increase system complexity and cost.

A standard digital parallel RGB interface configuration utilizes a 24-bit color depth, with 8 bits allocated per color channel. This configuration enables the display of approximately 16.7 million unique colors.

One critical part of the digital parallel RGB interface is the synchronization signals, specifically the horizontal sync (HSYNC) and vertical sync (VSYNC) signals. These signals, sent over separate lines, indicate when a line of pixels (or an entire frame) has ended, ensuring that the display and source are synchronized in their communication.

The back porch and front porch are essential elements in these synchronization signals. The term "porch" is derived from old television technology, where it was used to define periods in which the electron beam (rendering the image) would reset its position. In the digital parallel RGB interface, the back porch follows the HSYNC or VSYNC pulse and represents a period without data. The front porch, on the other hand, precedes the sync pulse and also represents a non-data period. These "porches" act as buffers, giving the display time to process the sync pulse and prepare for the following line or frame of data.

Finally, the clock line serves as the heartbeat of the digital parallel RGB interface. It sends out regular pulses to control the timing of the data transfer. This clock signal ensures that the source and display are in sync when transmitting and interpreting data. Each clock pulse represents an opportunity to transmit a new set of red, green, and blue values corresponding to a new pixel.

In this thesis, we will implement a reduced version of this interface. In particular, the interface will be implemented only in grayscale, thus with only one 8-bit channel (figure 2.7), during the first two clock cycles of every horizontal front porch, right after the end of the transmission of the pixels of a line, a CRC16 checksum is transmitted to validate the integrity of data.

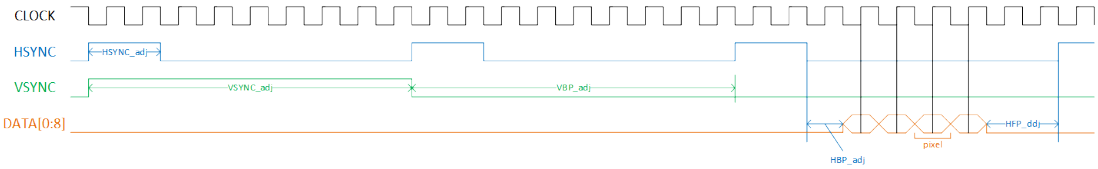


Figure 2.7: Digital Parallel RGB project implementation

3

Hardware and Tools

This chapter introduces and provides an understanding of the hardware components and tools utilized. The primary hardware and software toolset used in the course of this work consists of the AURIX™ TC375 Lite Kit Development Board, Saleae Logic Pro 16, Tektronix 6 Series Oscilloscope, and the Infineon LED driver characterization board.

3.1 Aurix™ TC375 Lite Kit Development Board

The Aurix™ TC375 Lite Kit Development Board, developed by Infineon Technologies, represents a cutting-edge integration of the 32-Bit Single-Chip Aurix™ TriCore based-Microcontroller, specifically the Aurix™ TC375. This development board has been meticulously designed to ensure versatility and is compatible with a diverse range of development tools. Among these tools are the Aurix™ Development Studio, which is Infineons proprietary Eclipse-based Integrated Development Environment (IDE), and the Eclipse-based free entry toolchain that has been collaboratively developed by Hightec, PLS, and Infineon.

Delving into the specifics of the board, it is equipped with an on-board Micro USB interface. Additionally, it can be externally powered, with a voltage range spanning from 5 V to 40 V. In terms of connectivity the majority of the Aurix™ pins are accessible via expansion connectors labeled X1 and X2 found on the bottom of the board. Furthermore, the board incorporates two Infineon Shield2Go connectors, Arduino-compatible connectors designed for 3.3 V, a mikroBUS™[1] connector, a DAP Debug connector, a CAN connector, and an RJ45 connector. Beyond these, the board also features a CAN transceiver and a Low Power 10/100 Mbps Ethernet Transceiver. User interactivity is facilitated through a single user push-button, complemented by three user LEDs. There's also a reset push-button and a potentiometer [1].

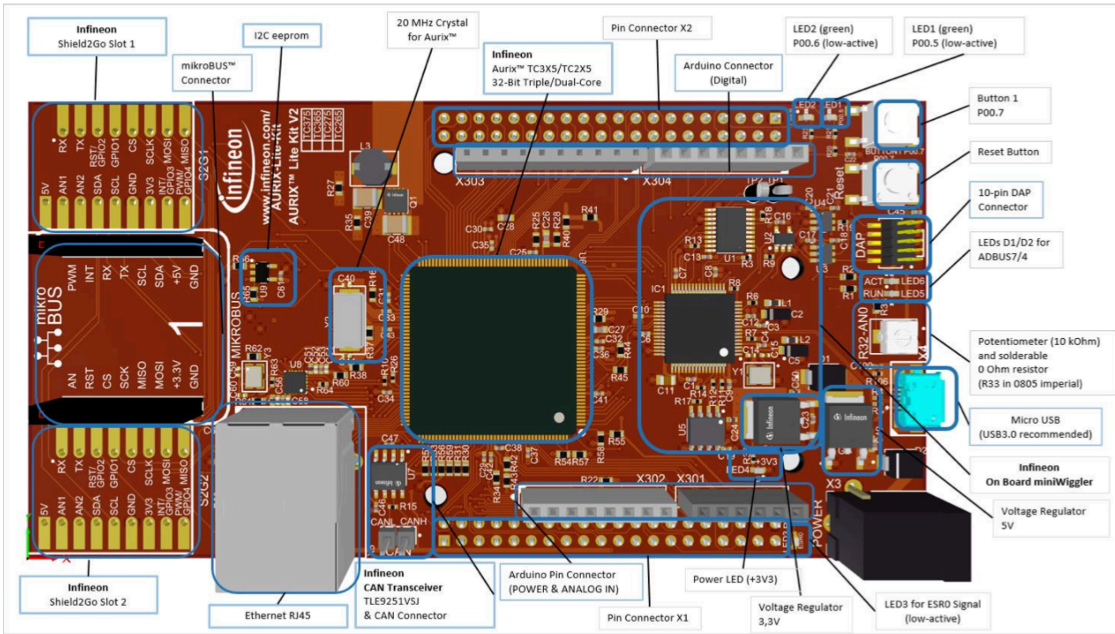


Figure 3.1: Aurix™ TC375 Lite Kit Development Board [1]

The development kit i powered by an Aurix™ TC375 SoC that houses three TriCore cores, each operating at a frequency of 300 MHz, proficient in executing both floating point and fixed point operations. The memory specifications include up to 6 MB of flash memory with Error-Correcting Code (ECC) protection and an SRAM capacity of up to 1.1 MB, also with ECC protection. The SoC's connectivity features encompass a 1 Gbit Ethernet, multiple CAN FD and ASCLIN ports, QSPI, IšC, SENT, PSI5, HSSL, MSC, and IšS emulation. Additionally, the SoC integrates the eVita full Hardware Security Module (HSM) with ECC256 and SHA2 encryption capabilities. The SoC is also compliant with the AUTOSAR 4.2 standard and is capable of operating on a single voltage supply of either 5 V or 3.3 V, with a standby mode controller further extending its

[1] mikroBUS is a standard specification by MikroElektronika that can be freely used by anyone following the guidelines. It includes SPI, I2C, UART, PWM, ADC, reset, interrupt, and power (3.3V and 5V) connections

functionality [2].

3.1.1 Toolchain and debugger

The Aurix™ Development Studio is identified as a comprehensive and complimentary Integrated Development Environment (IDE) specifically designed for the TriCore-based Aurix™ microcontroller family. This environment integrates the Eclipse IDE, a C-Compiler, a Multi-core Debugger, and Infineon's low-level driver, referred to as iLLD. The lack of time and code-size limitations facilitates developers in editing, compiling, and debugging application code without hindrance. The studio additionally provides numerous code example projects, illustrating its value in evaluating the architecture of the Aurix™ microcontroller family. A significant feature of the studio is the integration of the TASKING tools, including both the compiler and debugger, although it is important to note that these tools are designated solely for non-commercial applications. Regarding compatibility, the Aurix™ Development Studio is optimized for Microsoft Windows 10 as the host operating system.

3.2 Saleae 16 Pro Logic Analyzer and Logic 2 Software

During development, extensive use was made of a logical analyzer that is a testing instrument used to capture and display multiple signals from a digital system or digital circuit. It provided the visualization necessary to debug, verify correct operation of system. In particular it was used the Saleae Logic Pro 16 developed by Saleae a well-known brand that provides various logic analyzers to engineers, designers, and hobbyists. In the field of digital logic analysis, Saleae is a well-known brand that provides various logic analyzers to engineers, designers, and hobbyists. One of its flagship products is the Saleae Logic Pro 16, which offers high-performance digital logic analysis. In this chapter, we will introduce Saleae Logic Pro 16 and its accompanying software Logic 2. The device can analyze up to 16 digital channels simultaneously, making it suitable for complex digital designs. It uses a USB 3.0 interface to connect to the host computer, providing fast data transfer rates. It has a compact and portable design, making it easy to carry around. Key Features of Saleae Logic Pro 16 [7]:

- **High Sampling Rate:** provides a maximum sampling rate of 500MS/s [7], which is sufficient for most digital designs. The high sampling rate enables users to capture fast digital signals accurately.
- **Long Capture Time:** it has a maximum capture time of up to 4 billion samples, allowing users to capture long sequences of digital signals. This feature is particularly useful when analyzing digital signals that occur over a long period.
- **Digital Channels:** it supports up to 16 digital channels, enabling users to capture and analyze complex digital designs. The device provides enough channels to capture and analyze digital signals from multiple sources.
- **Triggering:** The device offers various triggering options such as edge triggering, pulse width triggering, and window triggering. These triggering options make it easier to capture specific events that occur in a digital signal.
- **Protocol Analysis:** Saleae Logic Pro 16 comes with built-in protocol analyzers for various digital protocols such as I2C, SPI, UART, CAN, LIN, and many more. The device can decode and display digital signals in a human-readable format, making it easier to analyze digital signals.

Saleae Logic Pro 16 comes with Logic 2 software, which is a modern and user-friendly interface for digital logic analysis. The software provides various features for analyzing digital signals, including waveform display, protocol decoding, and interactive triggering. Key Features of Logic 2 Software [7]:

- **Waveform Display:** The software provides a comprehensive waveform display that shows digital signals in a graphical format. The waveform display enables users to visualize the digital signals captured by the device.
- **Protocol Decoding:** Logic 2 software comes with built-in protocol decoders for various digital protocols such as I2C, SPI, UART, CAN, LIN, and also the possibility to create your own personalized. The software can decode and display digital signals in a human-readable format, making it easier to understand the captured data.
- **Interactive Triggering:** The software offers interactive triggering options that allow users to trigger specific events such as edge, pulse, or window.

The interactive triggering options make it easier to capture specific events in a digital signal.

- **Data Export:** Logic 2 software allows users to export captured data in various formats such as CSV, VCD, or MATLAB. This feature enables users to analyze captured data using other software tools, making it easier to integrate the data with other analysis tools.

One of the advantages of the Saleae Logic Pro 16 and Logic 2 software is their user-friendly interface. The software provides a comprehensive waveform display that shows digital signals in a graphical format. The waveform display enables users to visualize the digital signals captured by the device. Additionally, the software comes with built-in protocol decoders for various digital protocols, making it easier to understand the captured data.

3.3 Tektronix 6 Series Oscilloscope

The validation of generated signals is crucial for verifying the functionality of electronic systems. The Tektronix 6 Series Oscilloscope, specifically the 4-channel version, serves as a tool for this validation process. This oscilloscope incorporates FlexChannelő technology, enabling each of its 4 channels to process either one analog signal or eight digital logic signals. With a bandwidth range of 1 GHz to 10 GHz, and a sample rate of 25 GS/s for 4 channels, it is capable of capturing and analyzing a wide range of signal frequencies. The oscilloscope also has a record length of up to 1 Gpoints, a waveform capture rate of over 500,000 waveforms/s, and a 12-bit ADC extendable to 16-bits in High Res mode, indicating its high resolution and data handling capacity. It offers a variety of standard and optional analyses including frequency-domain analysis, advanced jitter, and eye diagram analysis, making it a versatile instrument for signal validation [6].

4 Methodology

4.1 Requirements analysis

The requirements analysis for the emulation of a parallel RGB interface using the Infineon Aurix™ microcontroller serves as a cornerstone for the project. The primary objective is to assess the feasibility of emulating this RGB interface with a targeted refresh rate of 100 Hz and the capability to transmit 8-bit grayscale images. The image data structure under consideration comprises 64 rows, each containing 256 pixels.

Key stakeholders play a significant role in shaping the project's requirements. The automotive industry, for instance, stands to benefit from this technology given the Aurix™ microcontroller's robust features and widespread application in automotive systems. Infineon Technologies, as the manufacturer of the microcontroller, also has a vested interest in the successful completion of this project. Furthermore, academic and research institutions could find this project valuable for further research in the field of microcontroller-based interface emulation.

Functional requirements are central to the project's success. The system must emulate the RGB interface effectively to transmit color information. Achieving a refresh rate of 100 Hz is crucial to have a fluid picture. The system should also be capable of transmitting 8-bit grayscale images, and a Cyclic Redundancy Check (CRC16) checksum should be implemented to ensure data integrity.

Non-functional requirements, although often overlooked, are equally important. Performance metrics such as system latency and throughput must be quantified and should meet or exceed industry standards. Scalability is another aspect that needs careful consideration. The system should be designed in such a way that it can accommodate future enhancements or additional functionalities.

Constraints and limitations are inevitable in any engineering project. The main constraint and factor that led to the decision to use the Aurix™ TC375 model was to select the model that had the minimum characteristics to achieve the objectives set, in order to minimize platform power consumption and, in the case of potential customer use, to ultimately reduce costs.

4.2 System design

This chapter presents a delineation of the software's architecture, focusing specifically on its division into two predominant areas: the GTM, which is responsible for signal generation, and the image preprocessing component, executed by the CPU.

In the architecture under study, the Aurix™ microcontroller's GTM serves as a central element, chosen for its capability to generate complex digital signals for real-time applications independent of the CPU. Within the GTM, the TOM module, which is tailored for PWM generation, was deployed to produce the clock signal, alongside the horizontal and vertical sync signals. Conversely, the generation of the 8 data line signals, which entail a higher degree of complexity, necessitated the use of the ATOM module. This is because it offers functionalities beyond those available in TOM, including advanced signal generation capabilities and interfacing with the GTM's internal Asynchronous Register Update (ARU) bus.

The ATOM's interface with the ARU allows for the incorporation of a First-In-First-Out (FIFO) module, facilitating automated data transfer to the ATOM registers. This setup aids in buffering data within the module, thereby streamlining data reception from the CPU. A schematic overview of the GTM module configuration is provided in the referenced figure 4.1.

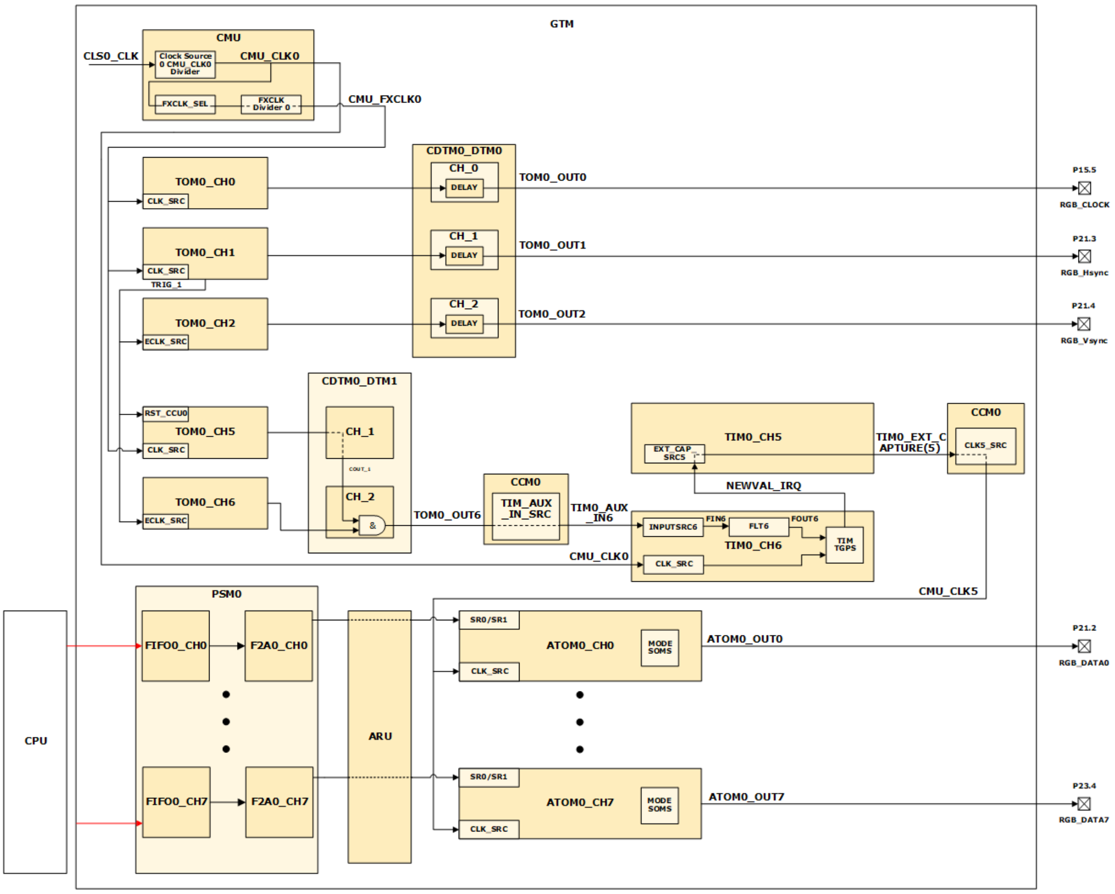


Figure 4.1: Schematic overview of the GTM module configuration

On the software side, the CPU firmware is bifurcated into two primary flows (see Figure 4.2). The principal flow manages tasks such as initializing the Light Emitting Diode (LED) driver, configuring and initializing the GTM for RGB interface emulation, and image preprocessing. Image preprocessing involves formatting the image data for efficient transmission to the GTM and performing Cyclic Redundancy Check 16 (CRC16) calculations. The processed data is then stored in a dedicated buffer within the microcontroller's RAM.

The secondary flow within the CPU is activated by an interrupt triggered by the GTM's FIFO module. This flow is responsible for transferring the preprocessed data from the RAM buffer to the GTM for transmission.

To optimize performance and enable concurrent operations, a double-buffer system is implemented in the CPU. This system ensures that while one buffer is being used for data transmission to the GTM, a second buffer is populated with the next image data set, thereby preparing it for immediate transmission once the first buffer is emptied. This dual-buffer architecture effectively minimizes latency and ensures a continuous data stream.

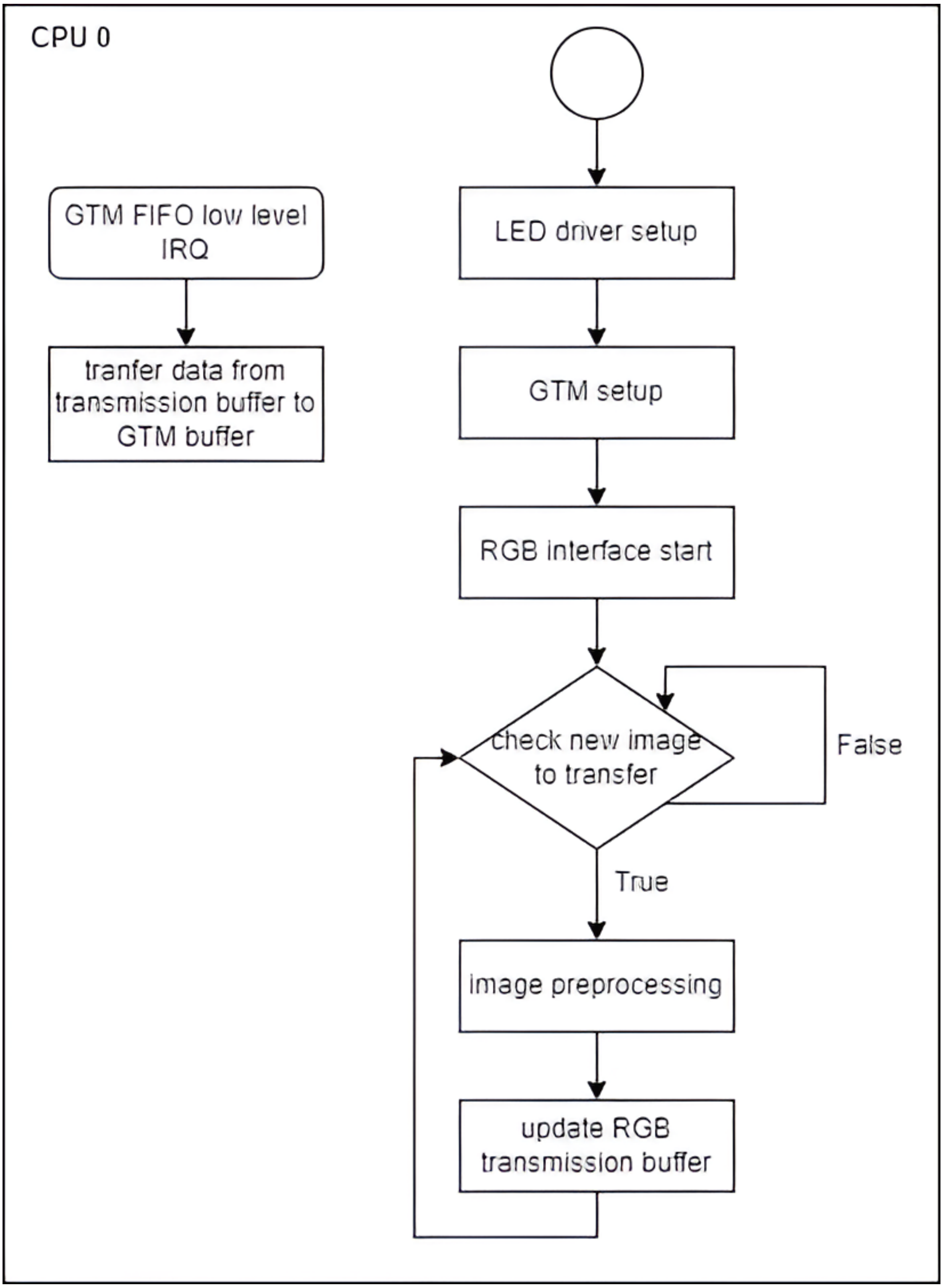

CPU 0
GTM FIFO low level IRQ
tranfer data from transmission buffer to GTM buffer
LED driver setup
GTM setup
RGB interface start
check new image to transfer
False
True
image preprocessing
update RGB transmission buffer

Figure 4.2: CPU software flow

4.3 Firmware implementation

In this section, we delve into the details of the firmware designed to achieve accurate emulation of the RGB interface. The discussion is structured around two primary components: the configuration of the Generic Timer Module (GTM) and the preprocessing activities conducted by the Central Processing Unit (CPU). Each of these components plays a critical role in the overall system architecture and contributes to the successful emulation of the RGB interface.

4.3.1 GTM configuration

The GTM module is activated with its operational clock set to its peak frequency of 200MHz. Through the Clock Management Unit (CMU), responsible for regulating the internal clocks within the GTM, both the CMU_CLK0 (used by the Dead Time Module (DTM) and the Timer Input Module (TIM)) and the CMU_FXCLK0 (specific to the Timer Output Module) are enabled. The CMU_-CLK0 is calibrated to its highest frequency, 200MHz, and to ensure consistency as a source for CMU_FXCLK0, it is configured to CMU_CLK0, effectively synchronizing these modules (see Figure 4.3).

Channels 0, 1, and 2 of the TOM have been employed to generate the Clock, horizontal sync, and vertical sync for the RGB interface, respectively. Channels 0 and 1 both utilize CMU_FXCLK0 as their base clock. However, for channel 2, this was infeasible due to the 16-bit size constraint of the compare registers, making it impossible to produce a sufficiently large PWM for the vertical sync signal. Therefore, the TRIG_1 pulse from TOM channel 1, designated for the horizontal sync signal, was exploited as the clock source. This signal is produced at the end of each PWM period, effectively creating a clock with the same frequency as the horizontal sync and enabling the generation of a sufficiently large PWM (see Figure 4.3).

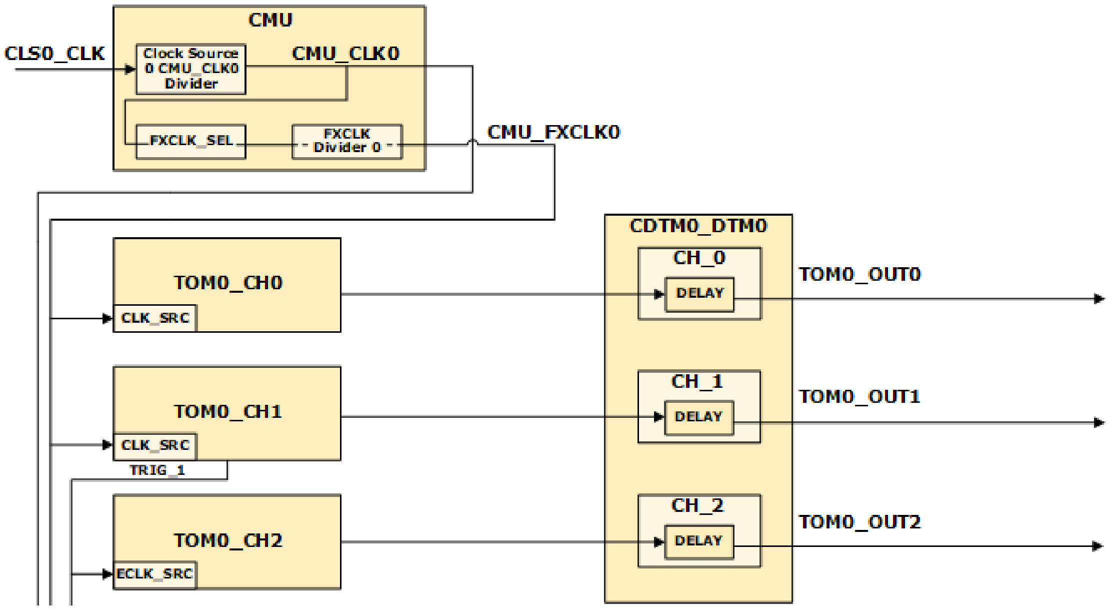


Figure 4.3: Clocks and RGB interface control signal configuration

To produce signals for the 8 data lines, one for each pixel bit, the ARU connected Timer Output Module (ATOM) was utilized with its eight channels (see Figure 4.4). Two intrinsic features of this module were leveraged. Firstly, its capability to operate in the Signal Output Mode Serial (SOMS) allows the ATOM channels to function as shift registers. With each clock pulse, they output a bit from their compare register. By loading the bits to be transmitted into each channel's registers and applying a common clock at the RGB interface frequency, parallel pixel transmission, as necessitated by the interface, is achieved. The second feature is the ATOM's connection to the GTM's internal bus (ARU), facilitating automatic pixel transfer using the FIFO module. This module will auto-load the upcoming transmission values into ATOM's shadow registers, while an interrupt signal is produced by the nearly empty FIFO, directing the CPU to replenish it with new data.

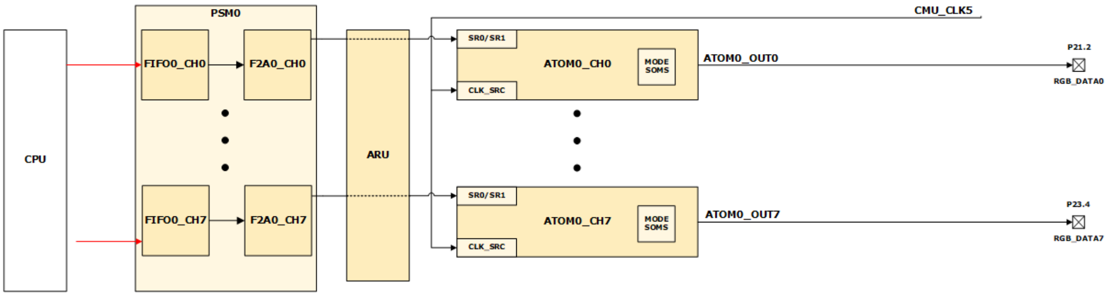


Figure 4.4: ATOM and FIFO configuration

The remaining challenge was generating a clock for the ATOM channels, which should only be active during data transmission periods and must match the RGB interface's clock frequency. The Timer Input Module was employed, leveraging its TIM Gated Periodic Sampling Mode (TGPS) capability (see Figure 4.5). In this mode, TIM samples a signal only when its received gating signal is 1. In this scenario, the provided signal for sampling is CMU_CLK0. By logically combining two PWMs generated with TOM channels 5 and 6, a gating signal was constructed to enable sampling solely during necessary data transmissions (see Figure 4.6). Upon enabling sampling, TIM was configured to count clock pulses in a manner that produces a NEWVAL_IRQ interrupt pulse with the same frequency as the RGB interface clock. Using the CCM0 module, these generated pulses via the interrupt were routed within the GTM as CMU_CLK5. Consequently, by designating CMU_CLK5 as the clock source for the ATOM channels used in data transmission, accurate data transmissions occur at the required periods and correct frequency.

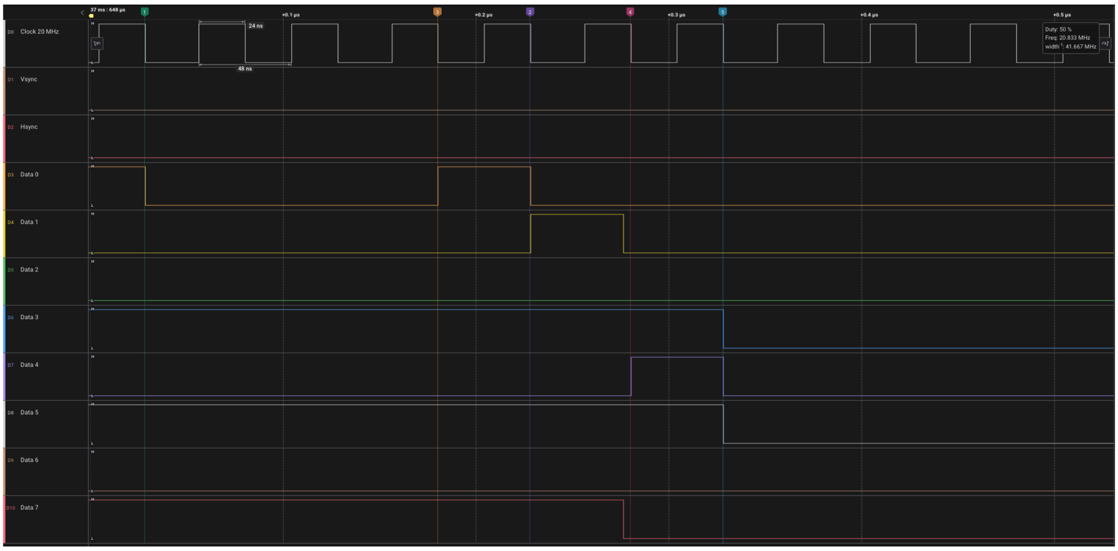


Figure 4.5: TOM gate signal and TIM TGPS configuration

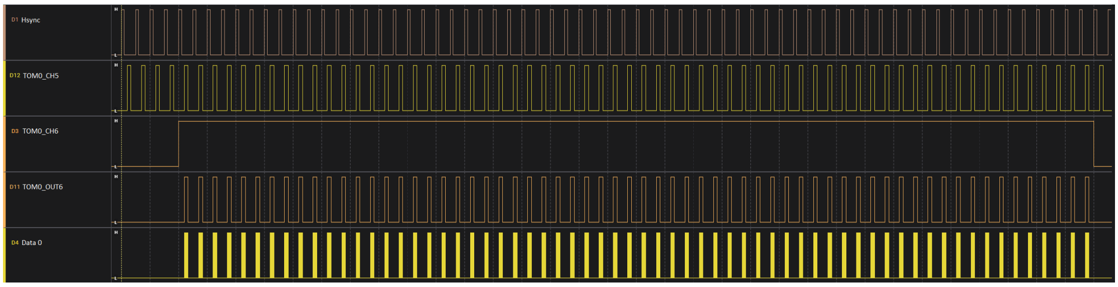
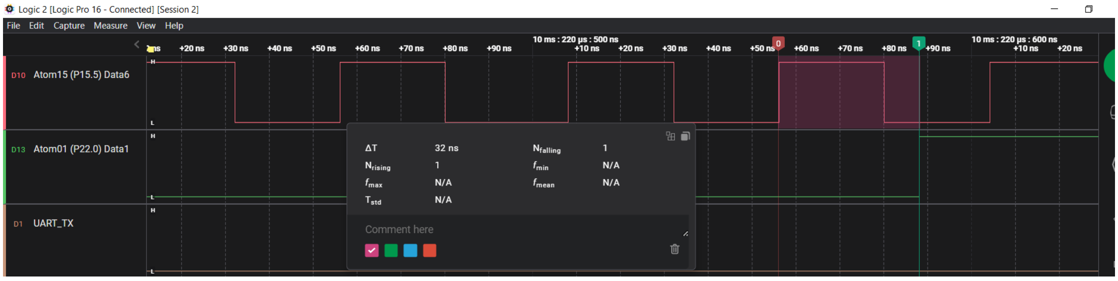


Figure 4.6: Gating signals

However, due to the multitude of modules employed, the generation of the ATOM clock introduced a significant delay, resulting in a 32 ns misalignment (see Figure 4.7) between the RGB interface clock and the data lines. This was addressed by utilizing the DTM to introduce a delay on the three control signals and by preemptively activating the gating signal input to TIM to achieve perfect alignment.


Logic 2 [Logic Pro 16 - Connected] [Session 2]
File Edit Capture Measure View Help
+20 ns +30 ns +40 ns +50 ns +60 ns +70 ns +80 ns +90 ns
10 ms : 220 µs : 500 ns
+10 ns +20 ns +30 ns +40 ns +50 ns +60 ns +70 ns +80 ns +90 ns
10 ms : 220 µs : 600 ns
+10 ns +20 ns
0
1
D10 Atom15 (P15.5) Data6
D13 Atom01 (P22.0) Data1
D1 UART_TX
H
L
Comment here

ΔT 32 ns
N_{rising} 1
f_{max} N/A
T_{std} N/A
$N_{falling}$ 1
f_{min} N/A
f_{mean} N/A

Figure 4.7: Logic analyzer capture of the delay between data lines and RGB clock

4.3.2 CPU preprocessing

As depicted in Figure 4.2, the firmware within the CPU encompasses two distinct code execution paths. The primary path initiates by initializing the LED driver for image transmission, contingent upon its connection to the microcontroller, and setting up the Generic Timer Module (GTM) for RGB interface emulation by configuring the requisite bit fields in the associated registers. Subsequently, two RAM buffers are allocated for storing data intended for GTM transmission.

Given that the ATOM transmission protocol mandates 24-bit wordsdetermined by the bit size of its transmission registersand that each word must contain only the n-th bit of the pixel corresponding to the data line associated with the ATOM channel, an image preprocessing phase is executed. Specifically, pixels are extracted from the 8-bit unsigned integer vector of the image in groups of eight. These eight pixels are treated as an 8x8-bit matrix, upon which a matrix transposition operation is performed. This results in the first byte, originally containing the first pixel of the eight, now holding the first-bit positions of all eight pixels, arranged in a manner suitable for transmission (see Figure 4.8.

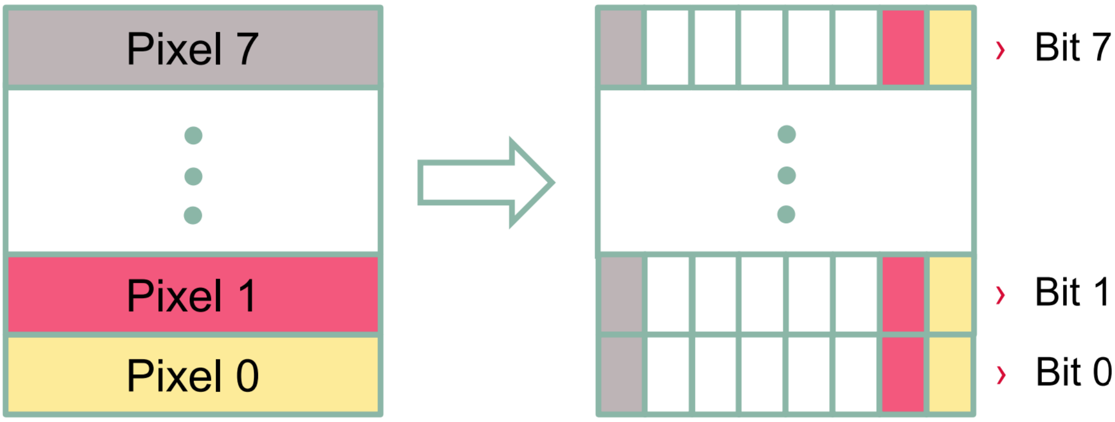


Figure 4.8: Result of the transpose on 8 pixels

These bytes are subsequently stored in the RAM buffer, formatted as 24-bit words. Concurrently, a 16-bit Cyclic Redundancy Check (CRC16) is computed for each row and appended to the transmission queue. This entire process is iteratively executed for each new image to be transmitted.

The secondary code execution path in the CPU firmware is triggered by the GTM FIFO interrupt. The corresponding interrupt service routine is responsible for transferring the preprocessed data from the RAM buffer to the GTM FIFO whenever the latter is nearing depletion.

5

Testing and validation

The objective of this chapter is to delineate the methodologies employed and the results obtained, specifically focusing on quantitative metrics, from a series of functional tests aimed at validating the RGB interface emulation implemented using an Aurix™ TC375 microcontroller. These tests were designed with two primary objectives. The first objective was to ascertain the correctness of the output signals generated by the interface. The second objective was to quantify any internal delays that may have been introduced by the Generic Timer Module (GTM) due to the internal routing required for clock generation. It is noteworthy to mention that the scope of this work did not extend to production-level validation due to constraints on resources and the exploratory nature of this research.

5.1 Preliminary Evaluation with Saleae Logic Analyzer

Initial tests for evaluating the output signal integrity and alignment of the RGB interface were conducted using a Saleae Logic Analyzer. A sampling rate of 500 MS/s was chosen, providing a temporal resolution of 2 nanoseconds. This resolution was instrumental for an initial scrutiny to confirm whether the timing specifications were adhered to see Figure 5.1). Tests were conducted across a range of clock frequencies up to 25 MHz. Within the tolerable capture error limits of the Saleae device, the RGB interface demonstrated correct operation and signal alignment (see Figure 5.2).

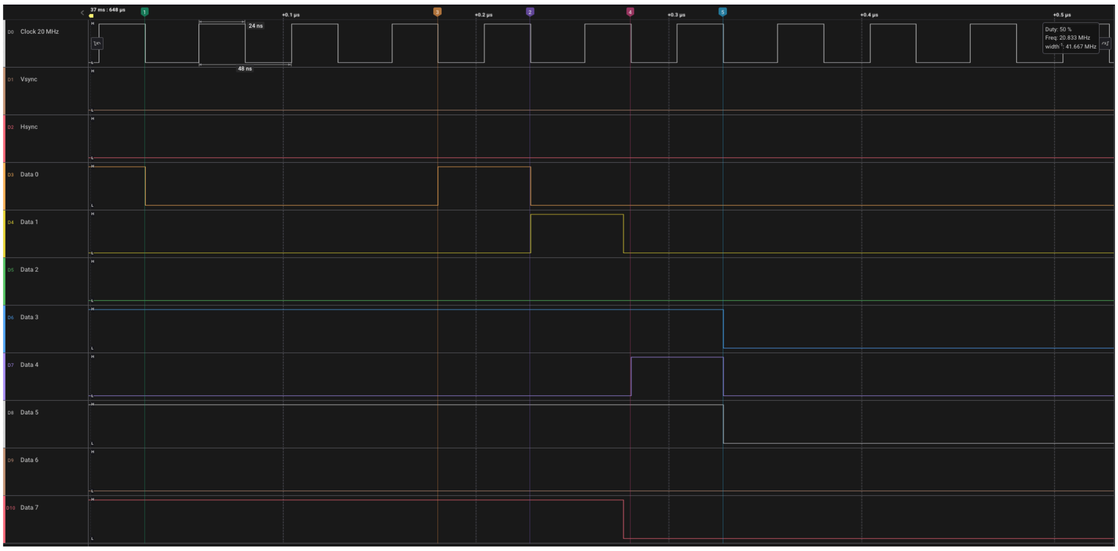


Figure 5.1: Logic analyzer signals alignment

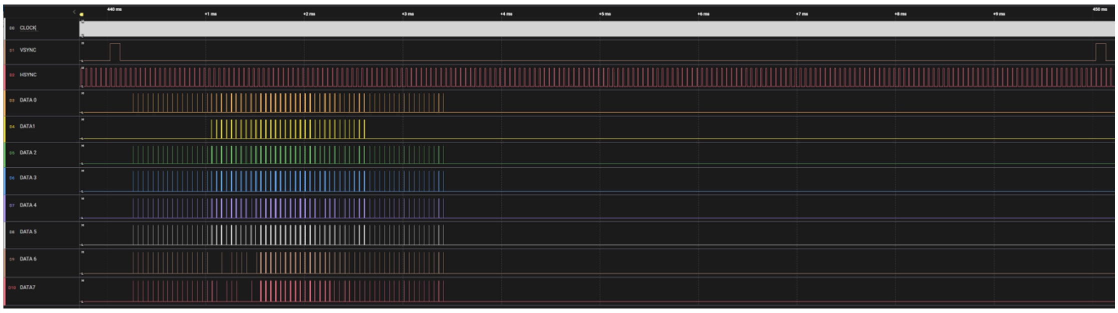
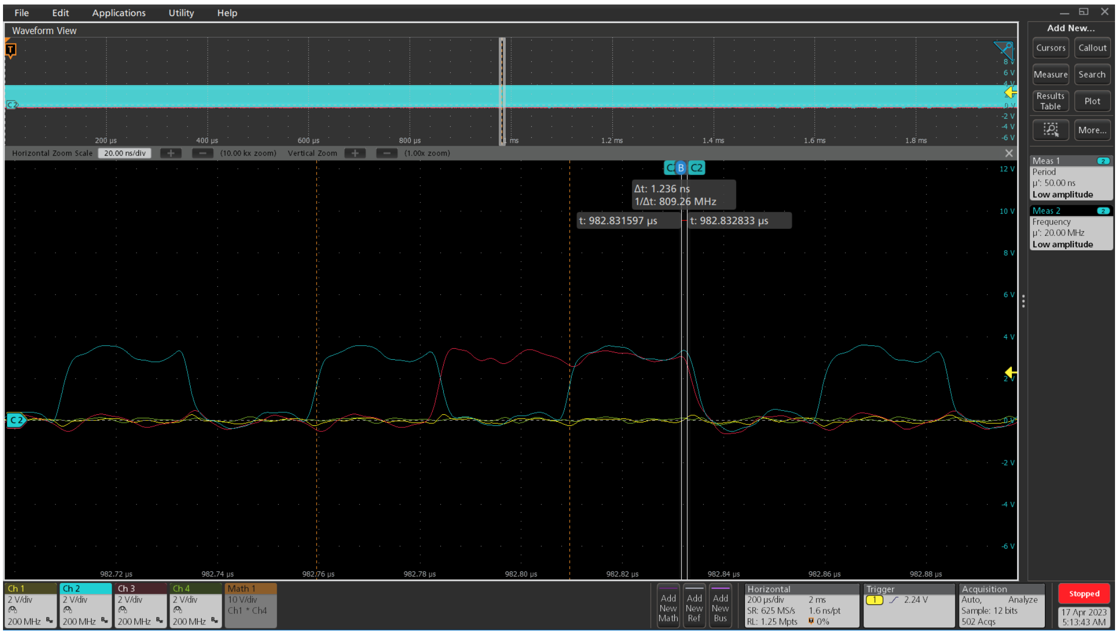


Figure 5.2: Logic analyzer capture of a transmitted image

5.2 Fine-Grained Analysis Using Tektronix 6 Series Oscilloscope

For a more detailed assessment of the alignment between clock and data signals, a Tektronix 6 Series Oscilloscope was employed. A considerably higher sampling rate of 200 MHz was utilized, and a rising edge trigger mechanism was set up to capture the relevant signals with higher precision. Upon analysis, it was observed that there was an approximate delay of 1.25 nanoseconds between the falling edges of the data and clock signals see Figure 5.3. Considering the inherent clock period of 2 nanoseconds for the GTM, this observed delay was at the limit of what could be feasibly compensated. The analysis was performed specifically at RGB interface clock frequencies of 10 MHz and 20 MHz.



Horizontal Zoom Scale 20.00 ns/div (10.00 kx zoom) Vertical Zoom (1.00x zoom)

Δt: 1.236 ns
1/Δt: 809.26 MHz

t: 982.831597 µs

t: 982.832833 µs

Meas 1: Period, µ': 50.00 ns, Low amplitude

Meas 2: Frequency, µ': 20.00 MHz, Low amplitude

Ch 1: 2 V/div, 200 MHz
Ch 2: 2 V/div, 200 MHz
Ch 3: 2 V/div, 200 MHz
Ch 4: 2 V/div, 200 MHz
Math 1: 10 V/div, Ch1 * Ch4

Horizontal: 200 µs/div, 2 ms, SR: 625 MS/s, 1.6 ns/pt, RL: 1.25 Mpts, 0%

Trigger: 1, 2.24 V

Acquisition: Auto, Analyze, Sample: 12 bits, 502 Acqs

17 Apr 2023
5:13:43 AM

Figure 5.3: Signals captured with the oscilloscope with RGB clock set at 20MHz (light blue clock, red data)

5.3 Final Validation Using Characterization Board

The terminal stage of validation was performed by interfacing the RGB module with a characterization board of an LED driver. The board was designed to accept this protocol as input, allowing for a physical verification of the image output. To facilitate this, a minor modification was made to the microcontroller's software. A subroutine was integrated into a secondary core to receive image frames from a UART interface (see Figure 5.4). This core then signaled the primary core, directing it to update the RGB interface's transmission buffer accordingly. The final test was conducted at a 20 kHz interface clock frequency, which is the minimum required to achieve the target image refresh rate of 100 Hz. Successful display of the transmitted images on the LED driver (see Figure 5.5) served as empirical validation that the microcontroller accurately emulated the RGB interface.

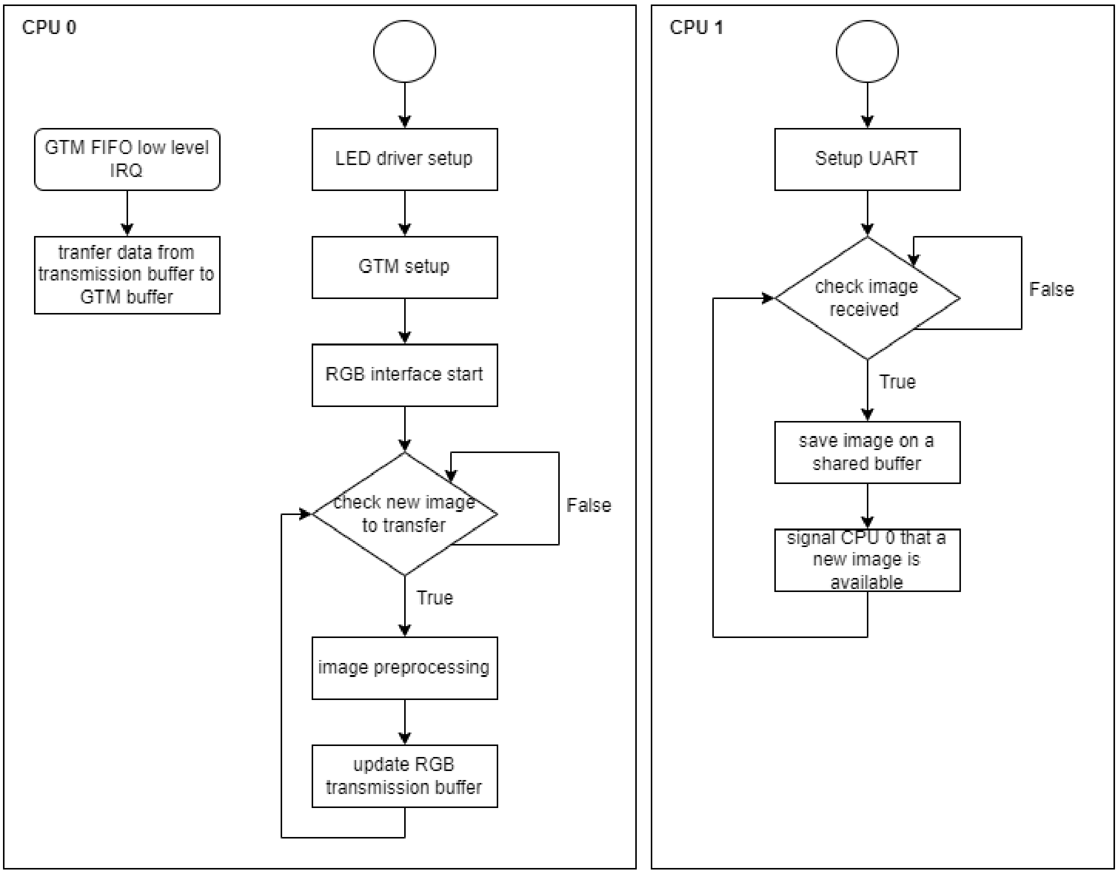

CPU 0
GTM FIFO low level IRQ
tranfer data from transmission buffer to GTM buffer
LED driver setup
GTM setup
RGB interface start
check new image to transfer
False
True
image preprocessing
update RGB transmission buffer
CPU 1
Setup UART
check image received
False
True
save image on a shared buffer
signal CPU 0 that a new image is available

Figure 5.4: Software flow used for testing with the Characterization Board

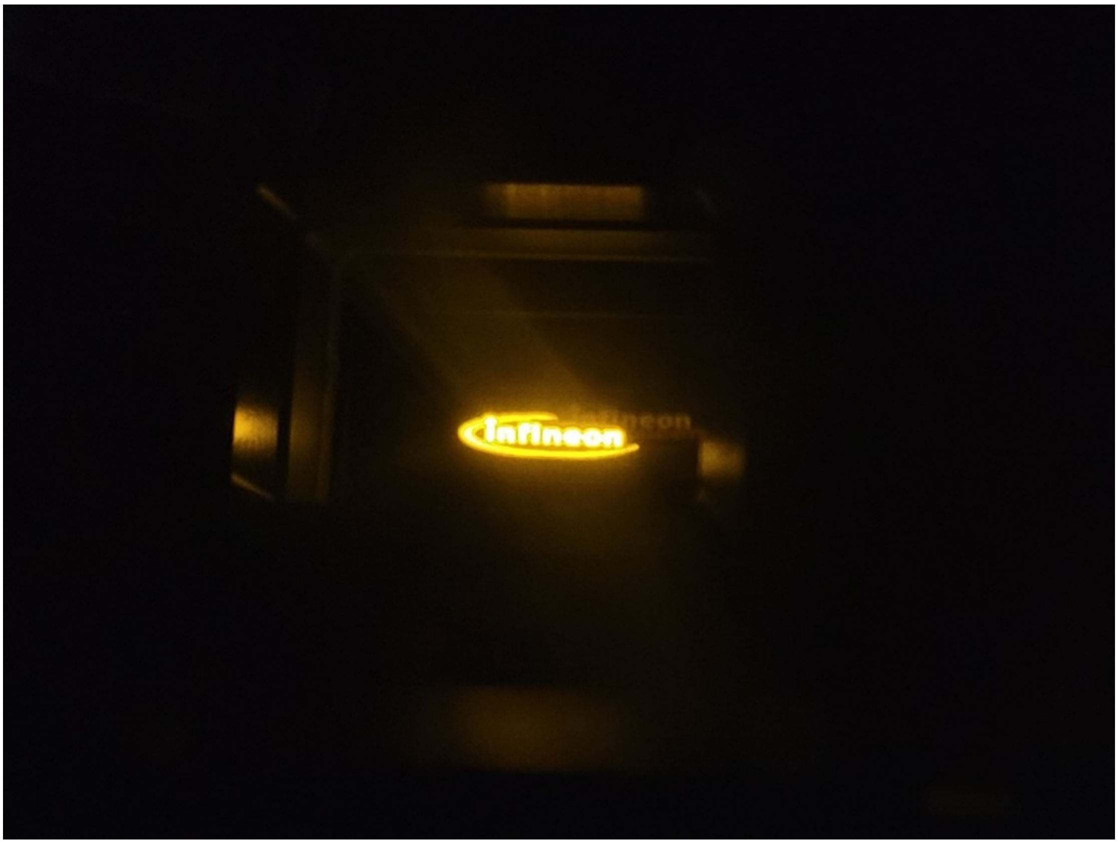


Figure 5.5: Image displayed by LED driver

6 Results and Discussion

6.1 Analysis of the results obtained

The results presented in previous chapter directly contribute to the primary objective of this thesis by demonstrating the Aurix™ TC375 microcontroller's feasibility for RGB interface emulation. The analysis of the results obtained from functional testing substantiates the effectiveness of the Aurix™ TC375 microcontroller in meeting this goal. Initial validation was performed using a Saleae Logic Analyzer operating at a sampling rate of 500 MS/s. This device was selected due to its high temporal resolution, essential for evaluating the intricate timing requirements of an RGB interface, and the ability to capture all the signal simultaneously. Within the capture error limits of the Saleae device, the timing and signal integrity were verified, as visually demonstrated in Figure 5.1, which affirms the timing congruence between the data and clock signals.

For a more nuanced understanding of signal alignment, a Tektronix 6 Series Oscilloscope was employed. This device was specifically chosen for its ability to provide a more granular analysis, vital for determining the alignment between clock and data signals. An observed delay of 1.25 ns between the falling edges of data and clock signals falls within the standard error margins, defined by the GTM module's inherent clock period of 2 ns (refer to Figure 5.3).

The integration of results from both the Saleae Logic Analyzer and the Tektronix 6 Series Oscilloscope provides a robust validation of the RGB interface's performance metrics. This comparative analysis indicates that the Tektronix 6

Series Oscilloscope offers finer temporal resolution, revealing details not captured by the Saleae Logic Analyzer, thereby affirming the necessity for multiple evaluation tools for comprehensive validation.

The terminal validation stage involved interfacing the RGB module with a characterization board of an LED driver. This phase confirmed the system's capability to transmit 8-bit grayscale images, comprising 64 rows and 256 pixels per row, at a targeted refresh rate of 100 Hz (refer to Figure 5.5).

Overall, the high-level temporal precision and signal alignment validated by these tests demonstrate that the Aurix TC375 microcontroller, particularly through its GTM module, fulfills the primary thesis objective of feasibly emulating an RGB interface. These results not only corroborate the research hypothesis but also open avenues for future exploration into more complex applications involving RGB interfaces higher resolution image transmissions.

6.2 Limitations

This section delineates the architectural constraints encountered during the implementation and testing of an RGB interface using the Aurix™ TC375 microcontroller and its Generic Timer Module (GTM). While the study achieved its primary objectives, it is critical to elucidate specific limitations that emerged due to the hardware constraints inherent in the chosen microcontroller and its modules.

The most substantial limitation is centered around the System Peripheral Bus (SPB), which is responsible for interfacing the TriCore CPUs with various peripherals, such as UARTs, ADCs, Ethernet, and CAN bus. When the RGB interface's transmission frequency was increased to 25MHz, the SPB was unable to facilitate a rapid enough data transfer to the GTM's internal First-In-First-Out (FIFO) buffer. This inefficiency led to buffer underflow and subsequent corruption of transmitted frames. It is important to highlight that the 20MHz tests were conducted in a relatively isolated environment with only a UART connection as the active peripheral. Therefore, the robustness of this implementation under more demanding conditions involving multiple active peripherals remains unverified.

Another constraint arises from the constrained size of the TOM registers. With a limitation of 16 bits, as opposed to the 24 bits offered by the ATOM, the minimum clock frequency of the interface is restricted to 813kHz. While potential modifications could be made like utilize triggers generated by the interface clock as its input source similarly to what has been done with vertical sync, this introduces a risk of misalignment in the generated control signals due to inherent GTM limitations as specified in the user manual [3] this happens the TOM channel used for horizontal sync is already generating a trigger signal and this could lead to the possibility to lose some trigger signals. Utilizing ATOM clusters could be a solution, but this would introduce additional complexities and latency due to routing signals across multiple clusters.

An additional constraint on the range of frequencies supported by the interface arises from the fact that the base frequency for clock signal generation is fixed at 200 MHz. To achieve a clock with a 50% duty cycle, the period divisor must be an even integer, thereby imposing a limitation on the available frequencies.

Lastly, scalability emerges as a constraint, particularly when considering an expansion in the number of data lines for future implementations. The existing architecture based on the TC37x series microcontrollers is ill-suited for such scalability due to the GTM's single FIFO module, which supports only eight channels. An upgrade to the TC39x series, which features two FIFO modules, would be imperative to address this limitation.

These constraints are rooted in the hardware architecture, and overcoming them would likely necessitate either a more complex system design or migration to a more advanced microcontroller series.

6.3 Practical applications

In the forthcoming chapter, we shall delve into the practical implementation of the research conducted in this thesis. Specifically, we will examine the integration of the Aurix™ TC375 microcontroller through the emulation of the RGB interface for the development of next-generation Advanced Front-Light Systems in automotive applications.

6.3.1 Advanced Front-Light Systems

Advanced Front-Light Systems (AFS) in automotive applications represent a significant leap in lighting technology, offering a range of features that go beyond the capabilities of traditional headlights. Utilizing high-resolution LED matrix technology, these systems, such as the one been developed by Porsche [8], can incorporate over 16,000 individually controllable micro-LEDs on a chip. This allows for a high-resolution light distribution that is up to twice as bright and covers a surface area four times larger than previous top-notch systems.

One of the key advantages of AFS is the highly flexible light distribution. The system can adapt to various driving conditions, thereby offering extremely homogeneous illumination. This is particularly beneficial for the driver, as it enhances visibility and reduces eye strain. The technology also includes innovative functions such as lane illumination, which brightens the vehicle's own lane significantly more than the surrounding areas. This function is activated exclusively on motorways or comparable roads and enables early detection of hazardous objects.

Another feature is the construction and narrow-lane light, which automatically adjusts the width of the light to match the vehicle's width, including mirrors, when construction zones or narrow areas are detected. This aids the driver in better assessing their position in the lane and enhances road safety. Additionally, the system includes adaptive motorway high-beam lights that optimally adapt the light distribution to the conditions on the motorway, preventing drivers in the oncoming lane from being dazzled (see Figure 6.1).

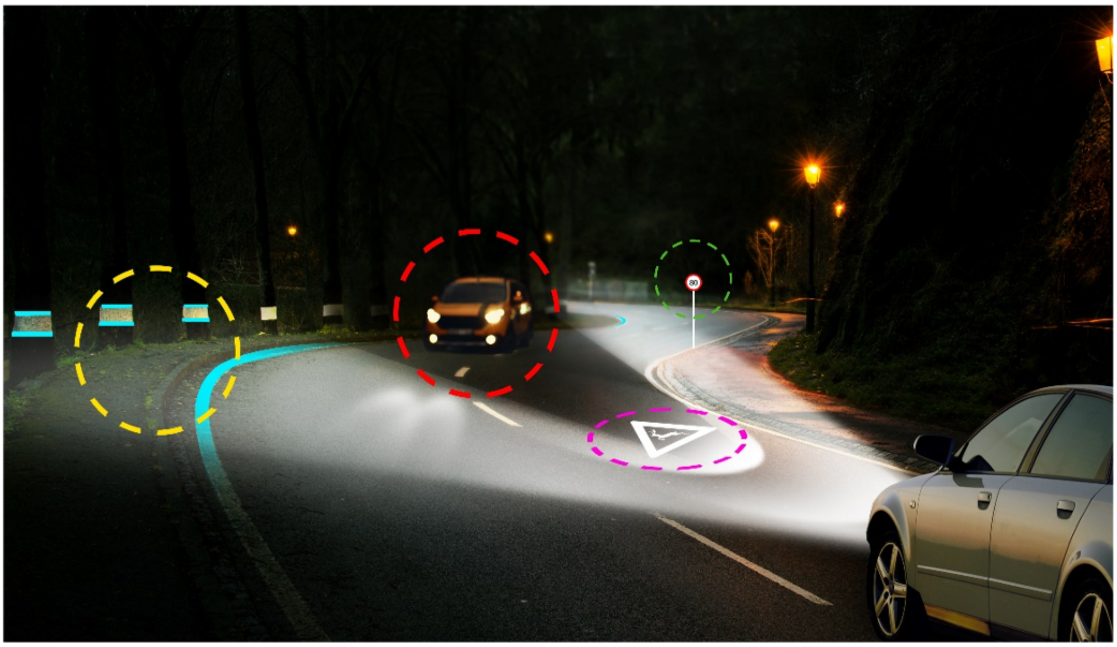


Figure 6.1: Example of operation of Advanced Front-Light Systems

AFS also incorporates a non-dazzling high-beam function. When the system's camera detects a vehicle ahead or an oncoming vehicle, the corresponding pixels of the HD matrix modules are switched off to prevent dazzling other drivers. The energy saved by this function is converted into additional functional HD light, optimizing the illumination of the non-dazzling high beam and improving the driver's visibility.

6.3.2 RGB emulation application

The integration of the Infineon Aurix™ TC375 microcontroller for RGB interface emulation in Advanced Front-Light Systems (AFS) presents a compelling case for enhancing automotive lighting technology. The RGB interface serves as a pivotal communication conduit between the microcontroller and the LED driver, enabling the Aurix™ TC375 to transmit specific image data for projection by the front-light system. This capability is instrumental for the dynamic features inherent in AFS, such as adaptive lane illumination and construction zone lighting. The emulation of the RGB interface allows for granular control over individual LED matrix pixels, thereby offering a high degree of customization in light distribution patterns.

When it comes to image storage and computational processing, the Aurix™ TC375 offers internal memory storage for less complex images. This is a practical solution for scenarios where the image data set is not extensive or computationally intensive. However, for more intricate image processing requirements, an external computational unit like the Infineon Traveo™ microcontroller can be deployed. The Traveo™, equipped with a GPU, can handle the computational burden of generating complex images. These processed images can then be transmitted to the Aurix™ TC375 via high-speed data interfaces such as Ethernet (see Figure 6.2), adding a layer of flexibility and capability to the AFS.

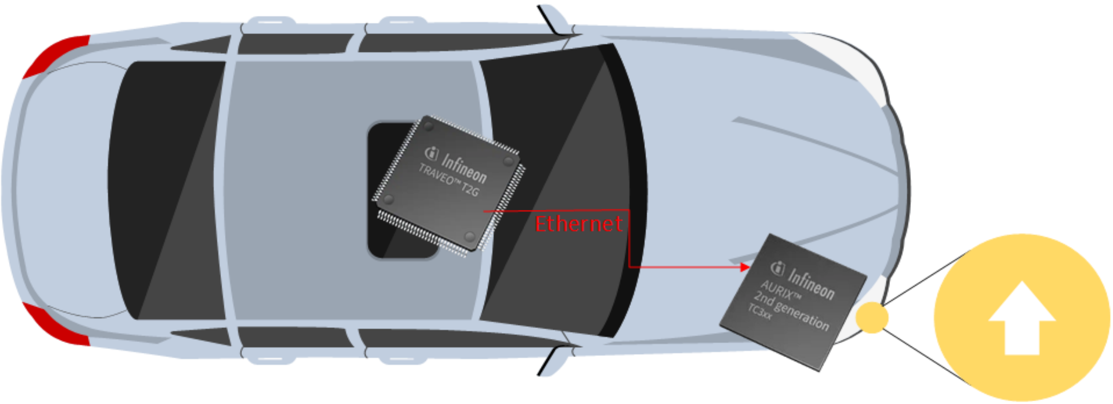


Figure 6.2: Example diagram of application with Traveo™ and Aurix™ connected by ethernet

The selection of the Aurix™ TC375 for this application is further justified by its stringent safety certifications and compatibility features. The microcontroller is ASIL-D certified, which is the highest level of Automotive Safety Integrity Level. This ensures that the microcontroller adheres to rigorous safety protocols, making it a reliable choice for automotive applications. Additionally, its compatibility with the AUTOSAR MCAL (Microcontroller Abstraction Layer) facilitates seamless integration into existing automotive systems and provides a standardized approach to hardware access at the microcontroller level.

In summary, the work outlined in the thesis on RGB interface emulation using the Aurix™ TC375 provides a robust technological framework for the development of next-generation Advanced Front-Light Systems. The approach combines precise control mechanisms, computational flexibility, and stringent safety standards, positioning it as an optimal solution for enhancing automotive lighting systems.

7

Conclusions and Future Works

This master's thesis presents a comprehensive investigation into the feasibility of emulating an RGB interface using the Aurix™ TC375 microcontroller, specifically leveraging its General Timer Module (GTM). The study successfully demonstrates that the Aurix™ TC375 microcontroller can emulate an RGB interface, fulfilling the primary objective of the research. The validation process involved multiple tools, including a Saleae Logic Analyzer and a Tektronix 6 Series Oscilloscope, which provided high-level temporal precision and signal alignment. The final validation stage involved interfacing the RGB module with a characterization board of an LED driver, confirming the system's capability to transmit 8-bit grayscale images at a targeted refresh rate of 100 Hz.

The research also identifies specific architectural constraints, particularly related to the System Peripheral Bus (SPB) and the the frequencies supported by the interface. These limitations could potentially affect the robustness of the implementation under more demanding conditions involving multiple active peripherals. Despite these constraints, the study opens avenues for future exploration into more complex applications involving RGB interfaces and higher-resolution image transmissions.

Possible future research directions may revolve around improving system scalability, the architecture currently in place is built upon the TC37x series of microcontrollers. One of the most salient limitations of this architecture is its inherent lack of scalability, particularly attributed to the Generic Timer Module's (GTM) incorporation of a single FIFO module. This limitation poses a significant constraint on the system's ability to adapt to more complex or demanding applications. As a potential solution, future research endeavors could focus on evaluating the TC39x series of microcontrollers, which are equipped with dual FIFO modules within the GTM. Such an architectural shift could provide a more flexible and scalable framework, thereby enabling the system to accommodate a broader range of applications and operational complexities.

Turning to the subject of interface support, the current system is designed with a focus on grayscale image processing. While this serves specific use-cases, it also restricts the system's applicability to a narrower domain. An intriguing avenue for future research would be to extend the system's capabilities to support full-color Red-Green-Blue (RGB) images. The incorporation of RGB support would significantly expand the system's applicability across various sectors.

Lastly, regarding system validation, the existing evaluation framework utilizes a Tektronix 6 Series Oscilloscope, which offers a high degree of temporal resolution, and a logic analyzer to evaluate the correctness of the generated signal. While these tools provide valuable insights into the system's short-term performance metrics, it may not be sufficient for assessing long-term operational stability. Therefore, future research should consider employing a diverse set of evaluation tools and methodologies. Such a comprehensive approach to validation would yield a more robust understanding of the system's performance under varying conditions and over extended periods, thereby contributing to its overall reliability and robustness.

In summary, the thesis provides a robust foundation for further research and development in the field of microcontroller-based interface emulation, particularly for automotive applications. The identified limitations and constraints offer specific directions for future work aimed at optimizing and extending the capabilities of the Aurix™ TC375 microcontroller for RGB interface emulation.

References

[1] Infineon Technologies AG. *AURIX TC375 lite Kit User Manual*. 2022. URL: https://www.infineon.com/dgdl/Infineon-AURIX_TC375_lite_Kit-UserManual-v02_20-EN.pdf?fileId=8ac78c8c7cdc391c017ce5193b575bd6.

[2] Infineon Technologies AG. *AURIX TC3xx User Manual part 1*. Infineon, 2021. URL: https://www.infineon.com/dgdl/Infineon-AURIX_TC3xx_Part1-UserManual-v02_00-EN.pdf?fileId=5546d462712ef9b701717d3605221d96.

[3] Infineon Technologies AG. *AURIX TC3xx User Manual part 2*. Infineon, 2021. URL: https://www.infineon.com/dgdl/Infineon-AURIX_TC3xx_Part2-UserManual-v02_00-EN.pdf?fileId=5546d462712ef9b701717d35f8541d94.

[4] Infineon Technologies AG. *TriCore TC1.6.2 core architecture manual Vol. 1*. Infineon, 2020. URL: https://www.infineon.com/dgdl/Infineon-AURIX_TC3xx_Architecture_vol1-UserManual-v01_00-EN.pdf?fileId=5546d46276fb756a01771bc4c2e33bdd.

[5] *Infineon Microcontroller*. URL: https://www.infineon.com/cms/en/product/microcontroller/ (visited on 10/07/2023).

[6] *Mixed Signal Oscilloscope Datasheet*. Tektronix. URL: https://www.tek.com/en/datasheet/6series-b-mso-mixed-signal-oscilloscope-datasheet (visited on 10/07/2023).

[7] *Saleae*. URL: https://www.saleae.com/ (visited on 10/07/2023).

[8] *Technology and function*. URL: https://media.porsche.com/mediakit/hd-matrix-workshop/en/texts/technik-und-funktion (visited on 10/07/2023).