# eTPU to GTM Migration

Parker Mosman, Hybrid Controls

Khaldoun Albarazi, STMicroelectronics

life.augmented

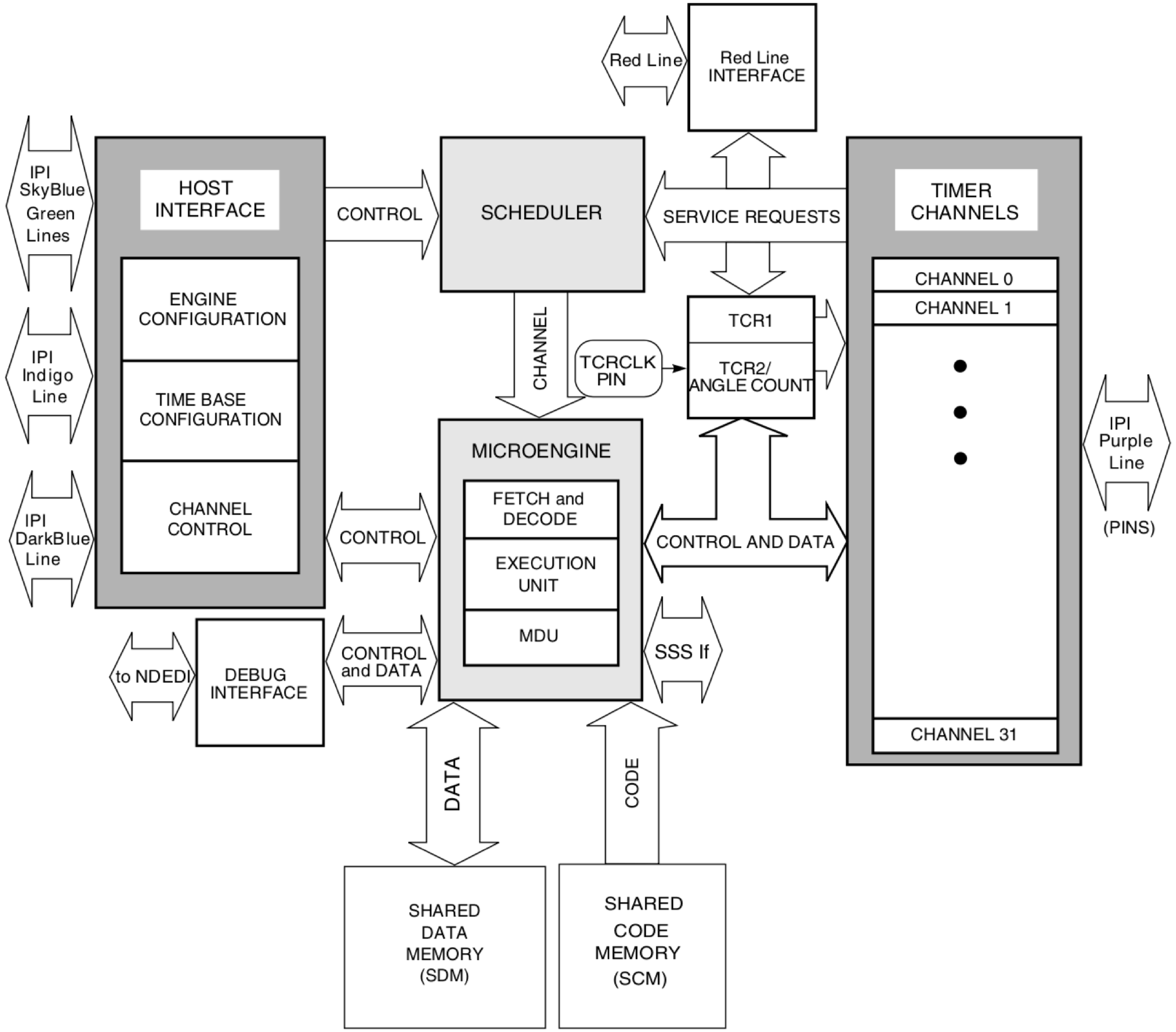
# Agenda



- eTPU / GTM Overview
- Feature Comparisons
  - Basic I/O
  - Angle Clock Generation
  - Mixed Domain Operations
- GTM Demonstration Material
  - Flywheel Simulator
  - Crank Synchronization
  - 8 Cylinder Fuel Injection
  - Angle Synchronous ADC
- ST Support Infrastructure

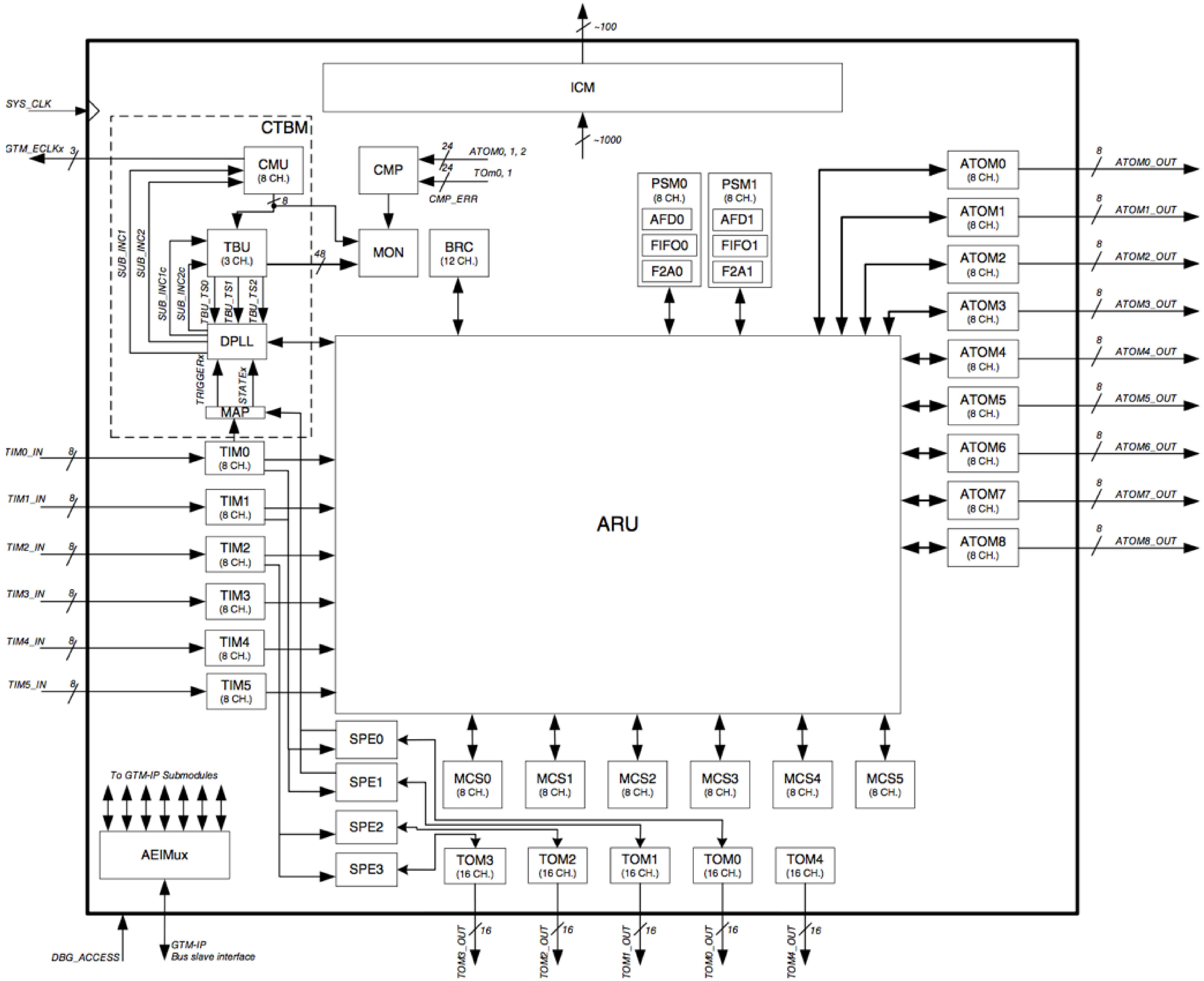
# eTPU Overview

- Generic timed input/output hardware connected to a powerful processing core
- Derives function and flexibility from user defined microcode
- Non-deterministic response: shared processing resources require worst-case latency analysis
- Requires user written microcode for any implementation
- Vendor specific IP



# GTM Overview

- Many specialized hardware blocks with dedicated functionalities
- Flexibility offered through hardware configuration (as memory mapped peripheral)
- Strongly deterministic response: ARU limits, but guarantees, worst case response time
- Many implementations do not require any GTM-specific microcode
  - Generic processing units are available for tasks which require this complexity/flexibility
- Vendor Independent IP



# Migration Overview



- Drastically different hardware = Drastically different solutions
  - eTPU implementations are software-centric
  - GTM implementations are typically hybrid solutions
- Classical porting from one to the other is not possible
  - Much larger effort than simply recompiling source code
  - Must begin at ECU hardware design time
- Some GTM applications will require no microcode development
  - Those that do, require substantially different software design to effectively leverage processing resources
- Strong determinism impacts your design approach

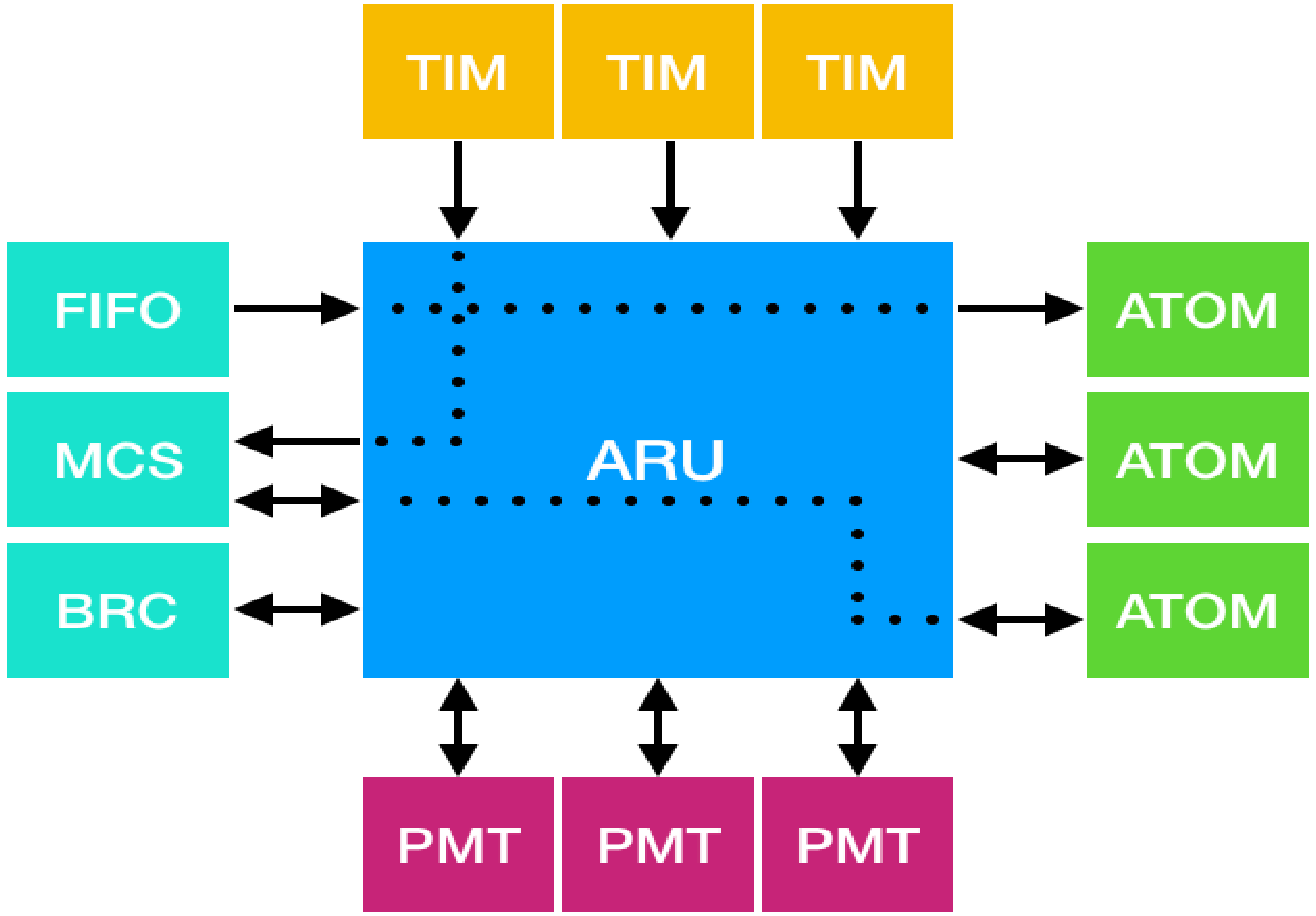
# GTM Hardware Overview

life.augmented

# Advanced Routing Unit (ARU)



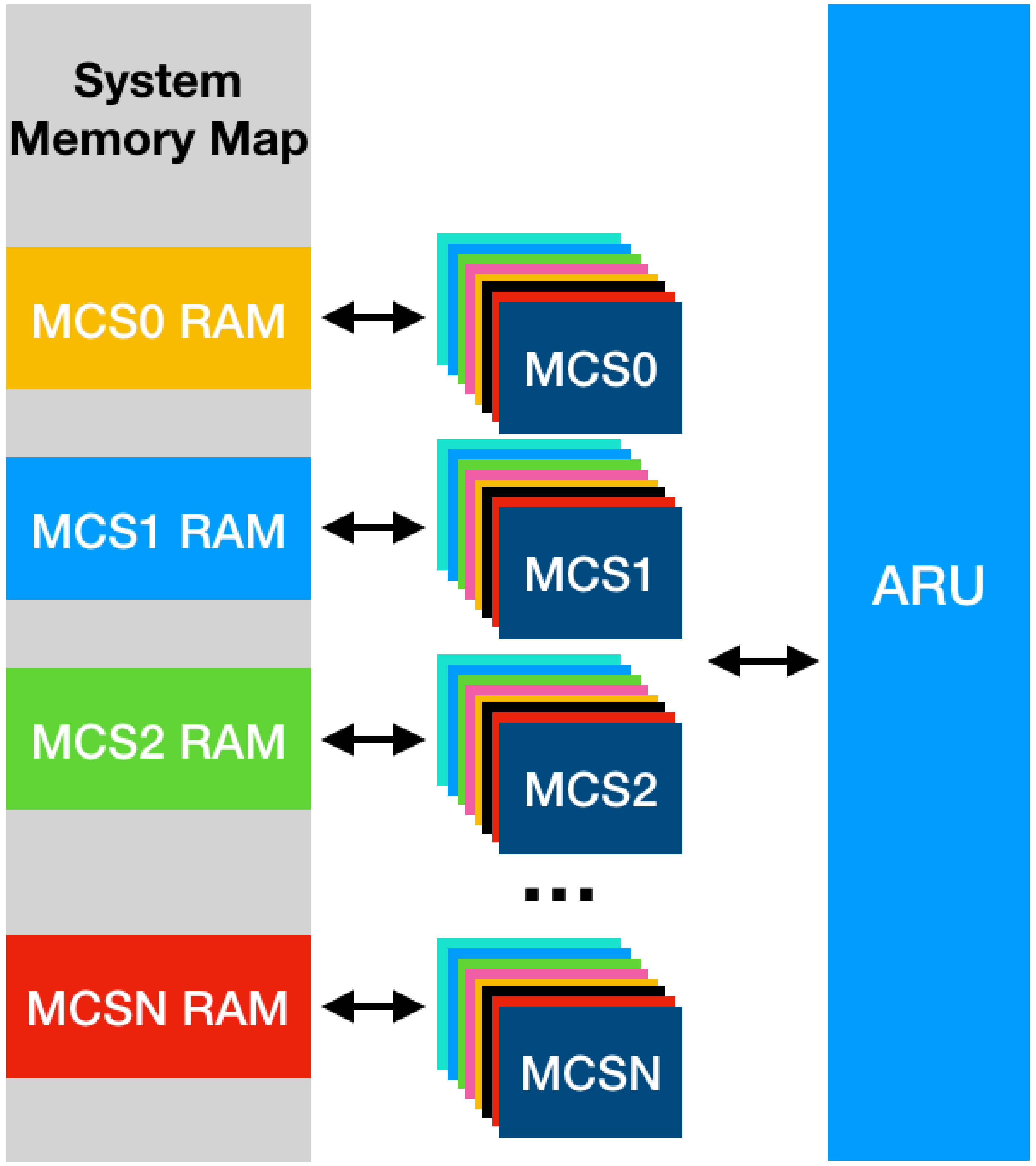
- A central piece of hardware which connects nearly every submodule to each other
  - Connections are arbitrary, and created by user configuration
  - All transfers are point-to-point
  - Facilitates 100% hardware solutions
- Uses a 53-bit data frame
  - Two 24-bit payloads and 5 control/status bits
  - Most submodules send/receive fixed format frames
- Operates in a strongly deterministic fashion
  - Each destination given equal time under all conditions
  - Thus, worst case latency is known at system design time
  - Eliminates stack-up analysis!



# Multi Channel Sequencer (MCS)



- Each MCS submodule has 8 channels (i.e. threads)
  - Threads share processing hardware, but have unique register files
  - Only one thread can execute at a time
  - Pseudo-parallel execution delivered at $1/9^{th}$ the system frequency (scheduling mode dependent)
- MCS submodules are 100% independent from each other
  - True parallel execution
  - Separate program and data memory
  - Can utilize unique scheduling modes
- Typical MCS applications:
  - Cam/crank synchronization
  - Fuel injection command generation
  - Spark ignition command generation
- Use of MCS is entirely optional!

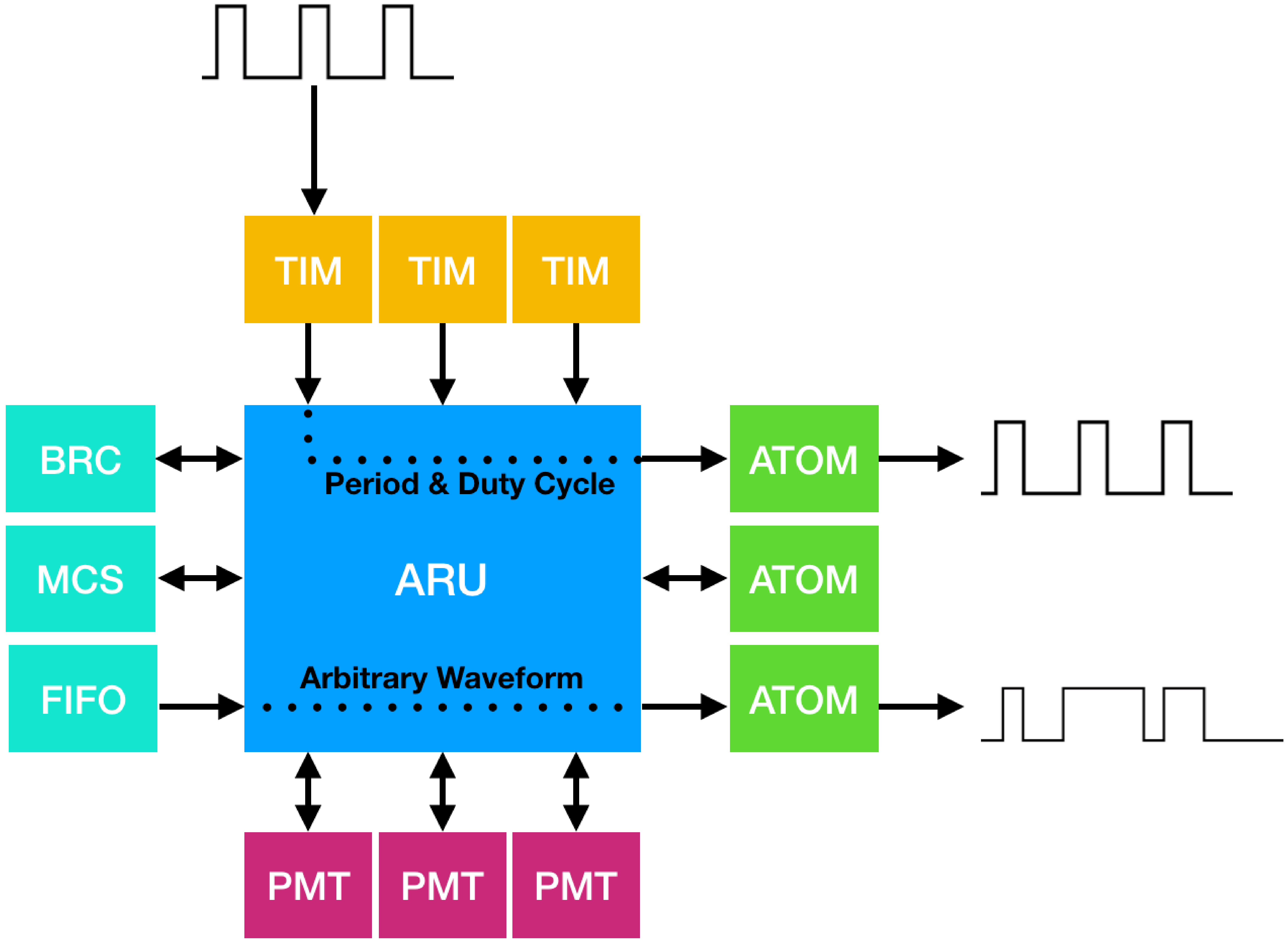


# Input and Output Modules (TIM/TOM/ATOM)



- Timer Input Module (TIM) offers a high degree of configurability
  - Operate in parallel to the CPU
  - Are ARU-connected with highly configurable payloads
  - Able to manage signal timeout and resynchronization autonomously
  - Can be controlled by MCS or host CPU
- Timer Output Module (TOM) is primarily used for simple PWM generation
  - Are not ARU-connected
  - Have only 16-bit resolution
  - Can be controlled by MCS (bus master interface only) or host CPU
- ARU-connected TOM (ATOM) is used for arbitrary waveform generation
  - Operates in mixed domain (e.g. time and angle)
  - Have 24-bit resolution and perform signed comparisons
  - Can source data from GTM FIFO to automatically generate complex waveforms in hardware
  - Can be controlled by MCS or host CPU

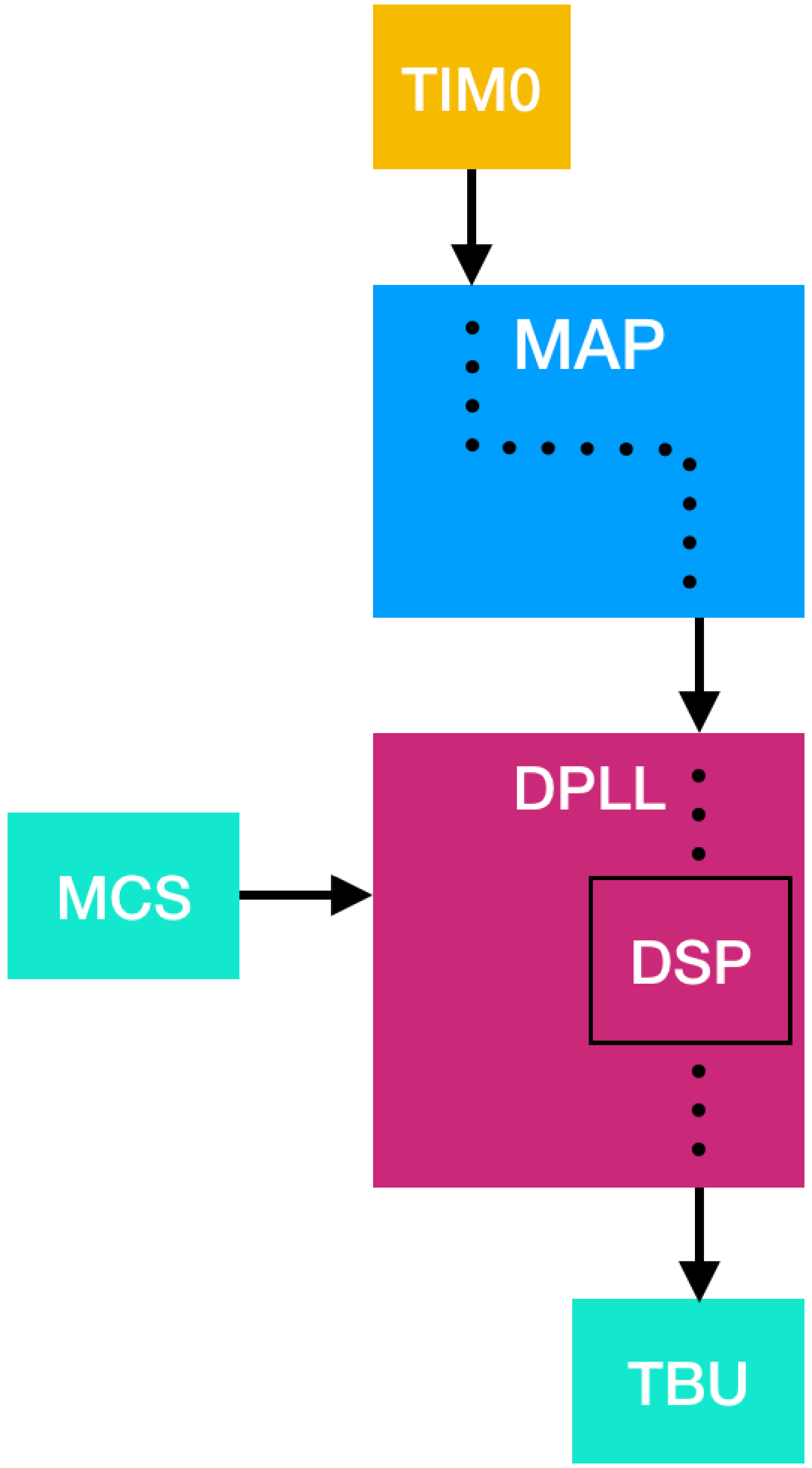
# Example: Hardware Driven Signal Generation



# Angle Clock Generation



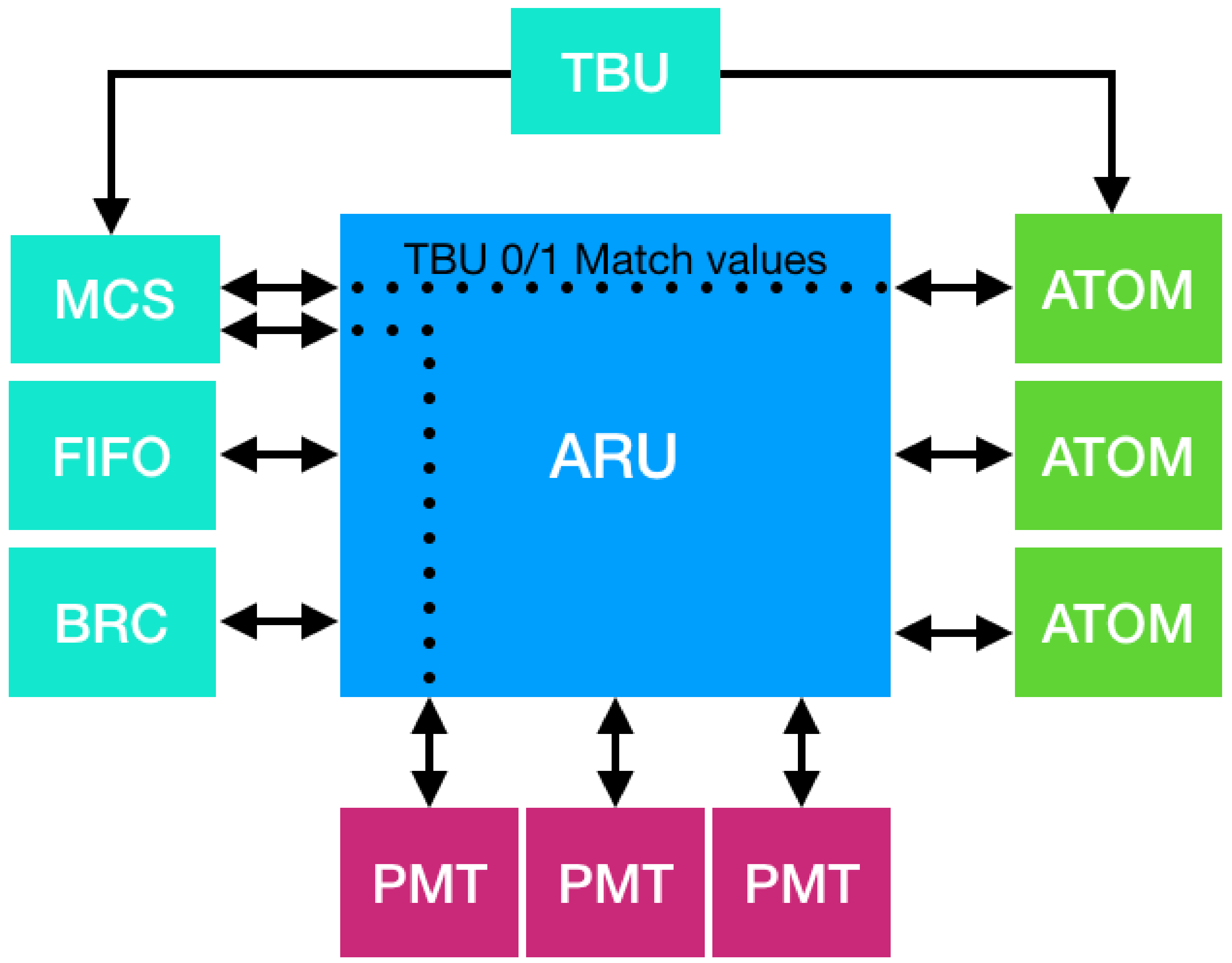
- TIM module 0 is special, and is utilized in generating the angle clock
  - The MAP module specifies which TIM0 channel should be used
- The DPLL module is comprised of custom DSPs and a fixed algorithm
  - Frequency multiplies the input of the TIM0 channel selected by MAP
  - Can achieve up to 1024 subdivisions of the input pattern
  - Uses this sub-increment clock to generate the angle clock
  - Operates in parallel to host CPU and MCS
  - DSP algorithm can be configured, but not modified
- The TBU generates up to 3 time bases globally available in the GTM
  - TBU_TS0 is always a free-running time-related time base
  - TBU_TS1 & 2 can be configured to be an angle-related time base (i.e. angle clock)
  - TBU_TS3 provides absolute position within working cycle to the host CPU



# Angle Synchronous Events



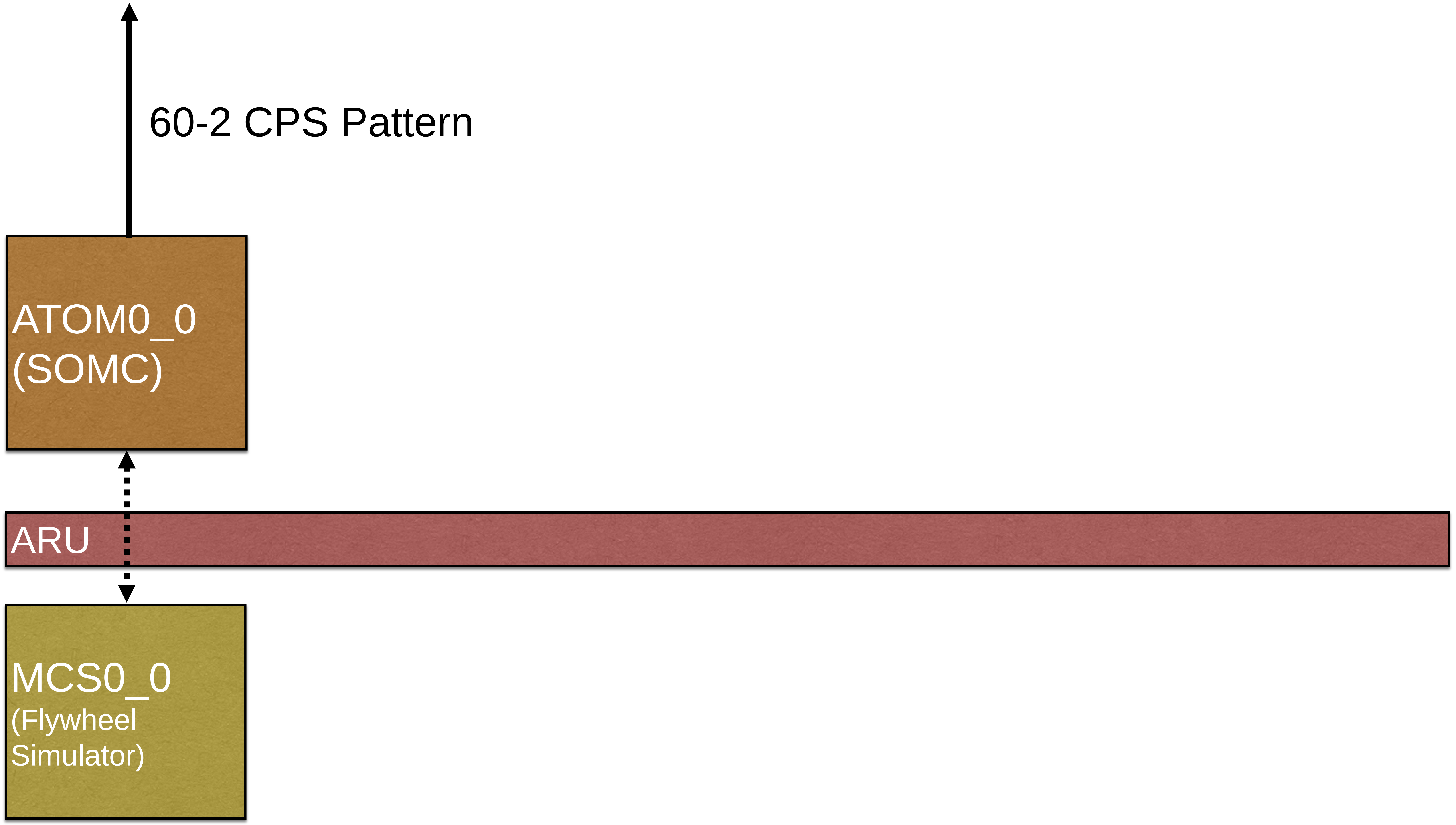
- ATOM and MCS have visibility to the TBU, and therefore can take action in the angle domain
- ATOM performs output transitions based on time and/or angle
  - Configurable matching patterns (e.g. time only, angle only, time then angle, etc.)
  - Performs signed comparisons for “in the past” detection
- MCS can observe TBU_TS1/2
  - Can issue commands in angle domain (e.g. ATOM match, host interrupt, etc.)
- Position Minus Time (PMT/DPLL action channels)
  - DPLL can be used to perform mixed domain predictions
    - MCS or host CPU specify an angle and a time duration before that angle
    - PMT computes the angle clock value of the event and time until the event will occurs
  - Compensates for engine dynamics, and is enhanced with every new timing event
  - Use case: Computing start time for ignition coil dwell



# ST & HyCon Training Exercise Overview

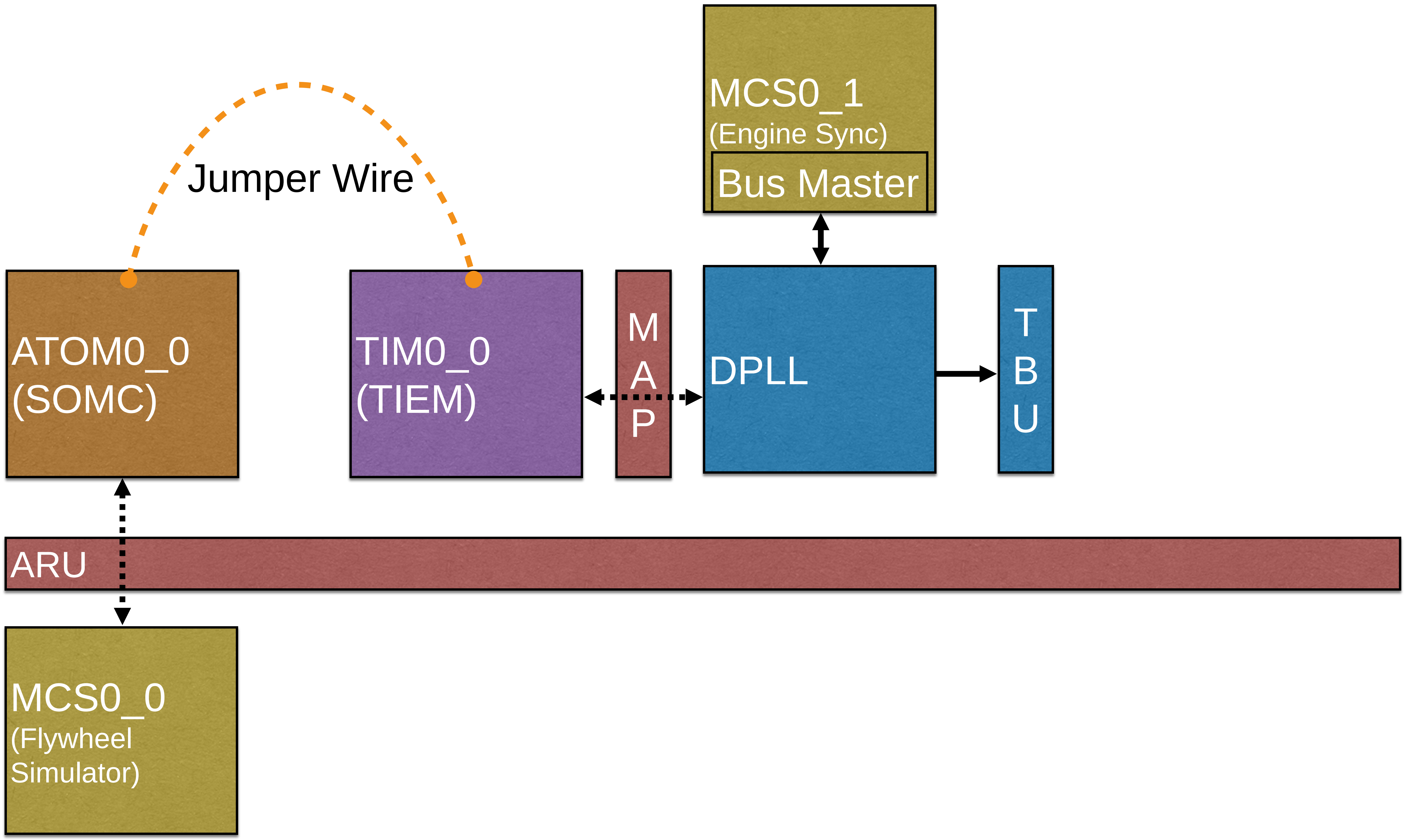
life.augmented

# Exercise 3: Crank Position Sensor Simulation



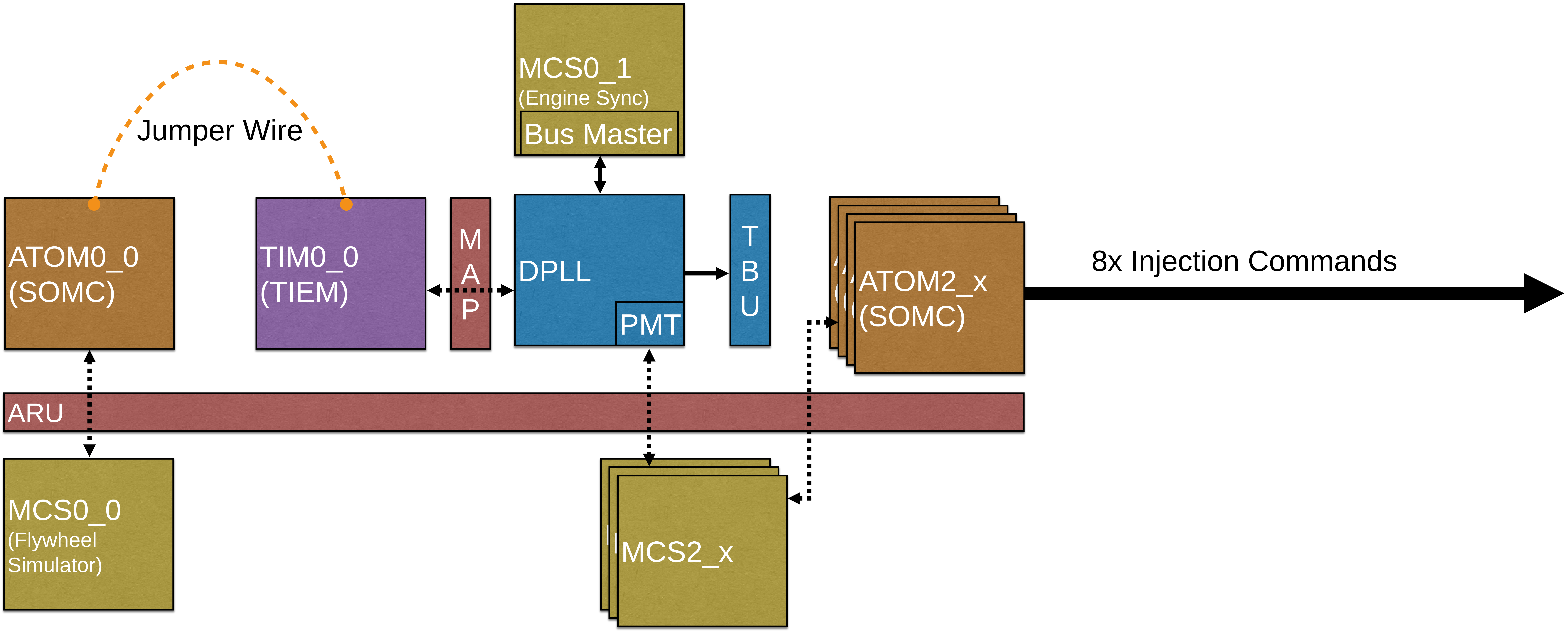
# Exercise 4: Crank Synchronization





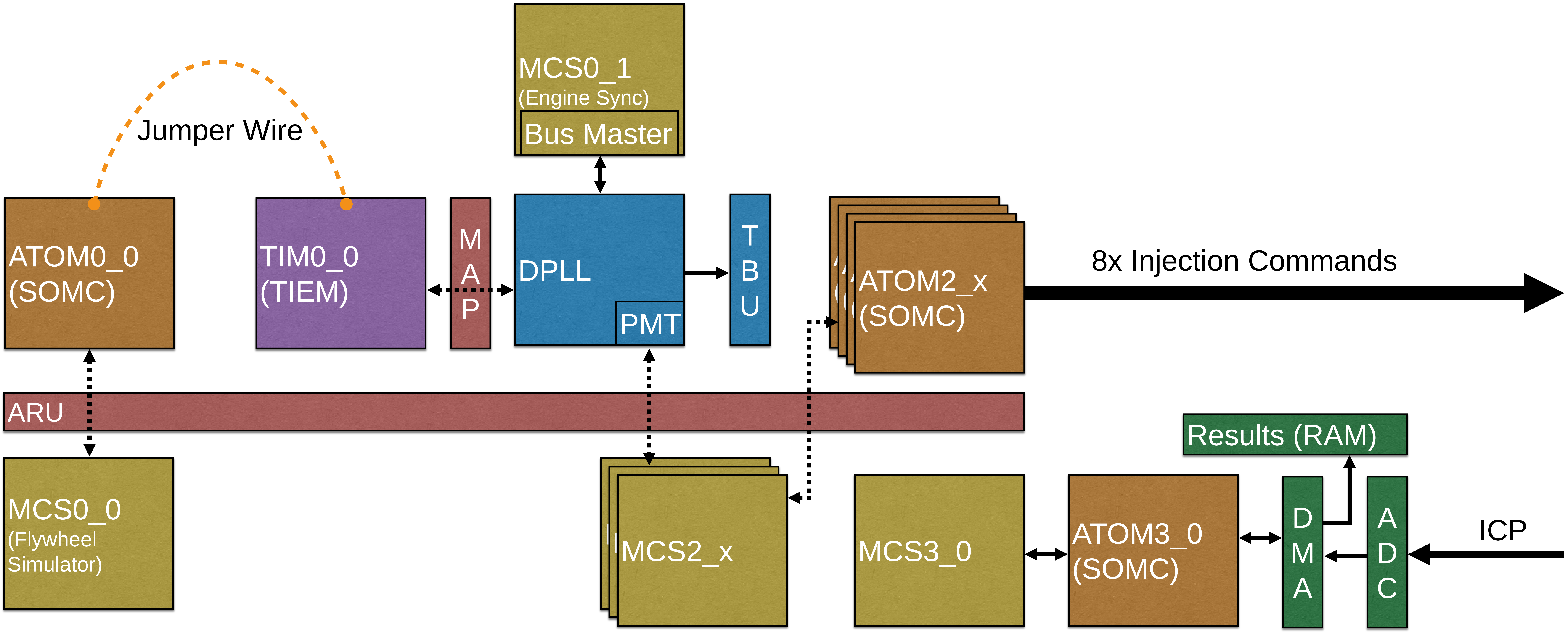
# Exercise 5: Gasoline Direct Injection





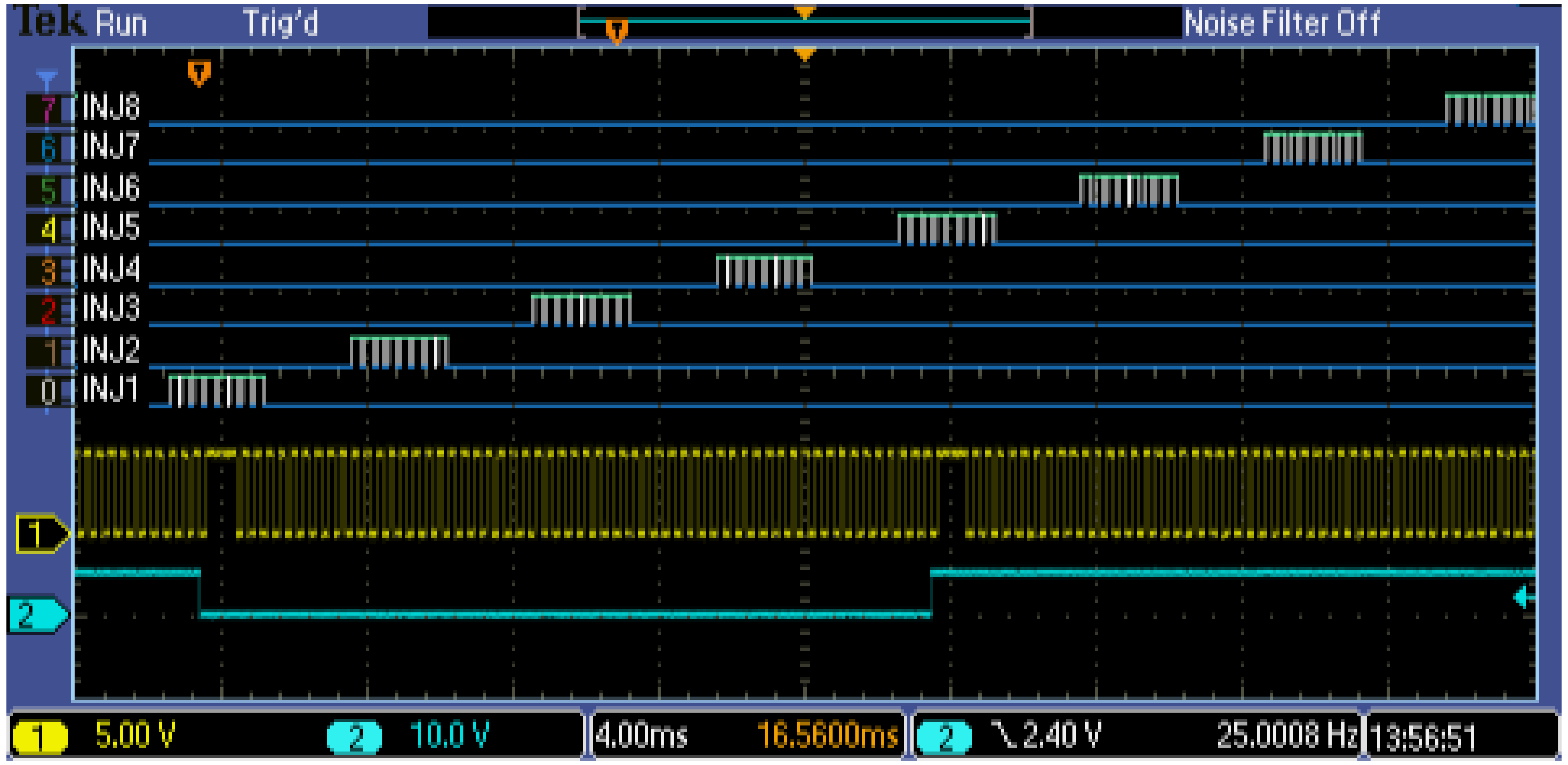
# Exercise 6: Angle Synchronous ADC



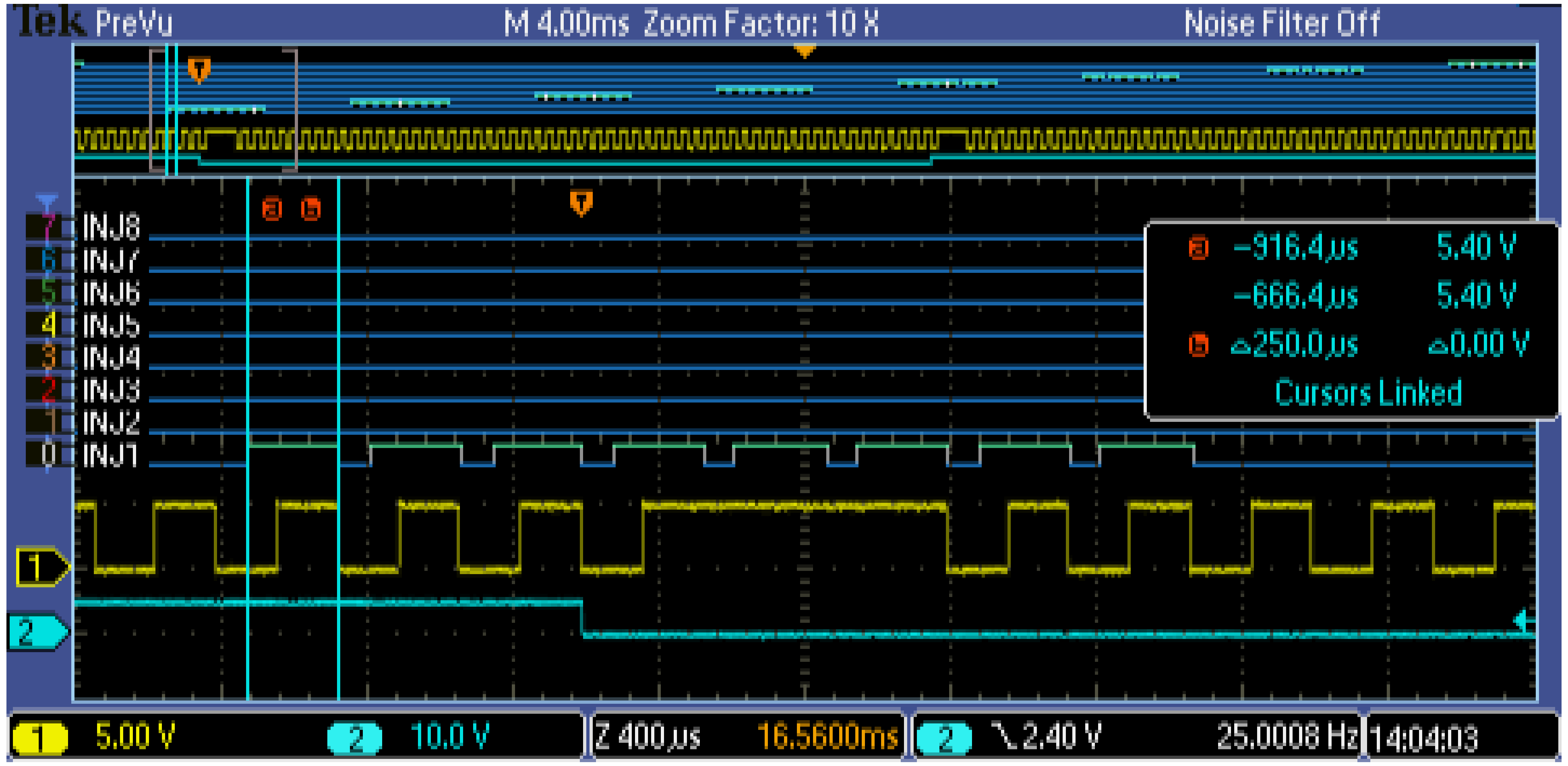


# Exercise 5 Results: Gasoline Direct Injection





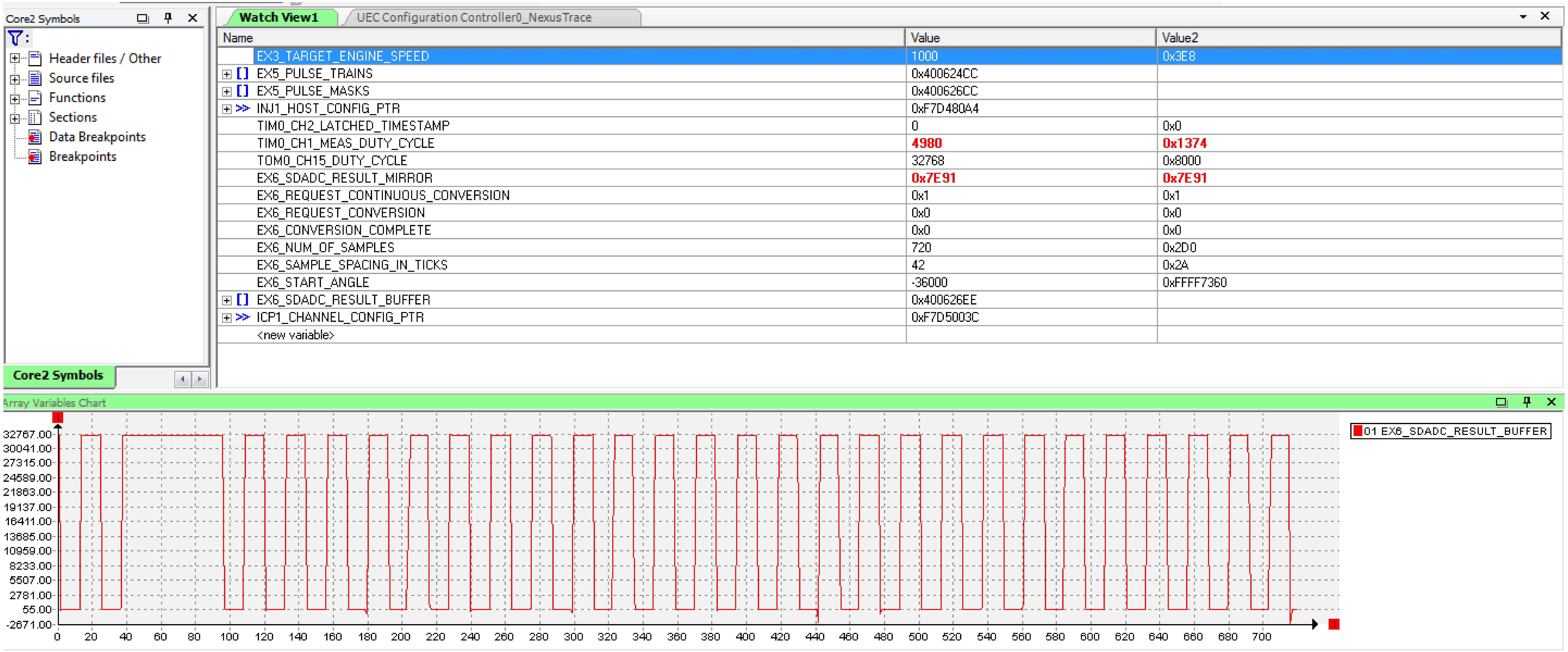

Ex5, 3000rpm, 8cyl x 8inj x 250usec




Ex5, 3000 rpm, 1cyl x 8inj x 250usec

life.augmented

# Exercise 6 Results: Angle Synchronous ADC




| Name | Value | Value2 |
|---|---|---|
| EX3_TARGET_ENGINE_SPEED | 1000 | 0x3E8 |
| [] EX5_PULSE_TRAINS | 0x400624CC | |
| [] EX5_PULSE_MASKS | 0x400626CC | |
| >> INJ1_HOST_CONFIG_PTR | 0xF7D480A4 | |
| TIM0_CH2_LATCHED_TIMESTAMP | 0 | 0x0 |
| TIM0_CH1_MEAS_DUTY_CYCLE | **4980** | **0x1374** |
| TOM0_CH15_DUTY_CYCLE | 32768 | 0x8000 |
| EX6_SDADC_RESULT_MIRROR | **0x7E91** | **0x7E91** |
| EX6_REQUEST_CONTINUOUS_CONVERSION | 0x1 | 0x1 |
| EX6_REQUEST_CONVERSION | 0x0 | 0x0 |
| EX6_CONVERSION_COMPLETE | 0x0 | 0x0 |
| EX6_NUM_OF_SAMPLES | 720 | 0x2D0 |
| EX6_SAMPLE_SPACING_IN_TICKS | 42 | 0x2A |
| EX6_START_ANGLE | -36000 | 0xFFFF7360 |
| [] EX6_SDADC_RESULT_BUFFER | 0x400626EE | |
| >> ICP1_CHANNEL_CONFIG_PTR | 0xF7D5003C | |
| <new variable> | | |

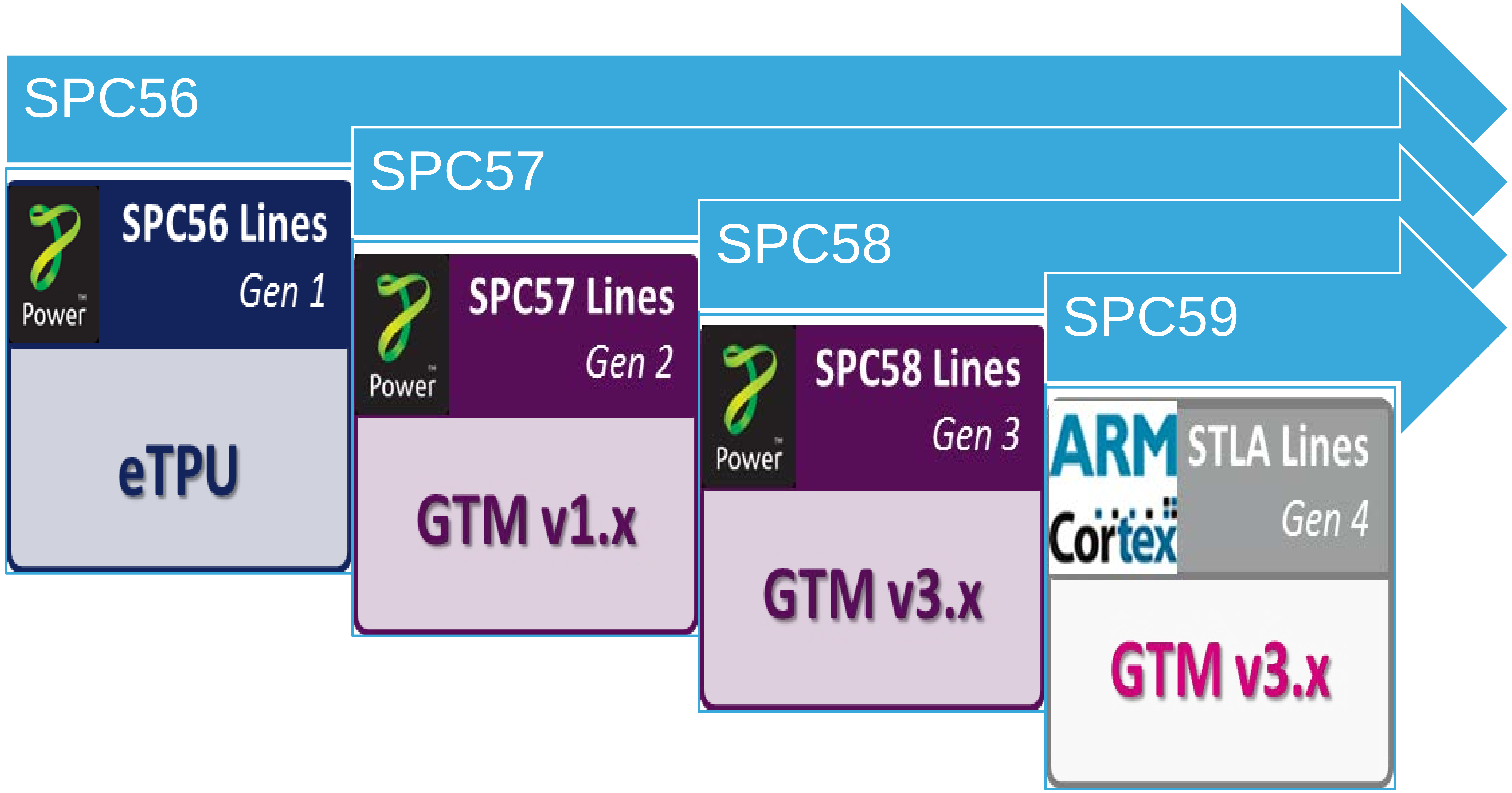
life.augmented

HYCON

# ST & HyCon Have You Covered!

# 32bit Automotive Microcontroller Roadmap

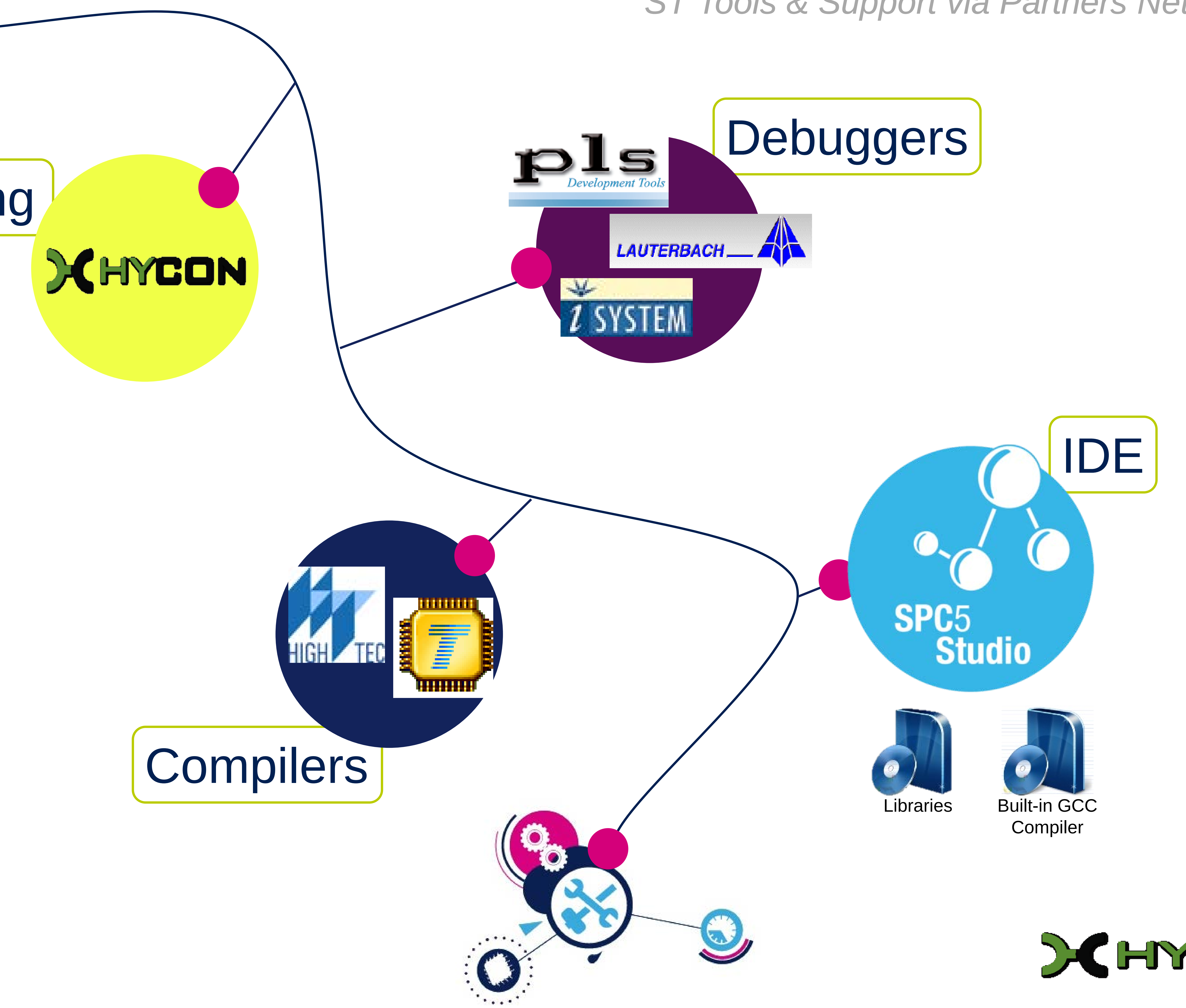
*Complex Timer Integration*



# GTM Support

*ST Tools & Support via Partners Network*

**Complete Development Ecosystem**




- ST has partnered with HyCon to develop a full spectrum GTM training course based on ST SPC58x microcontrollers
- The training includes a set of hands-on labs that emulates real, practical use cases of the GTM
- Covered topics include:
  - Top level architecture and submodules
  - Assembly & C MCS programming
  - DPLL configuration and runtime management
  - Safety critical functions
  - Advance debugging and data tracing

# Thank You!


life.augmented

HYCON