Lecture 16
Wrapup + Microprocessors

CE346 – Microprocessor System Design

Branden Ghena – Fall 2022

Some slides borrowed from:
Josiah Hester (Northwestern), Prabal Dutta (UC Berkeley)

Administrivia

- Bonus office hours on Friday
  - Friday 10-12 and 1-3pm in CG50

- Tuesday and Thursday after Thanksgiving will be office hours too
  - Likely in my office: Mudd 3305

Today's Goals

- Discuss remaining parts of the Microbit and nRF52833
    - Realize that we've covered almost everything on the system!!
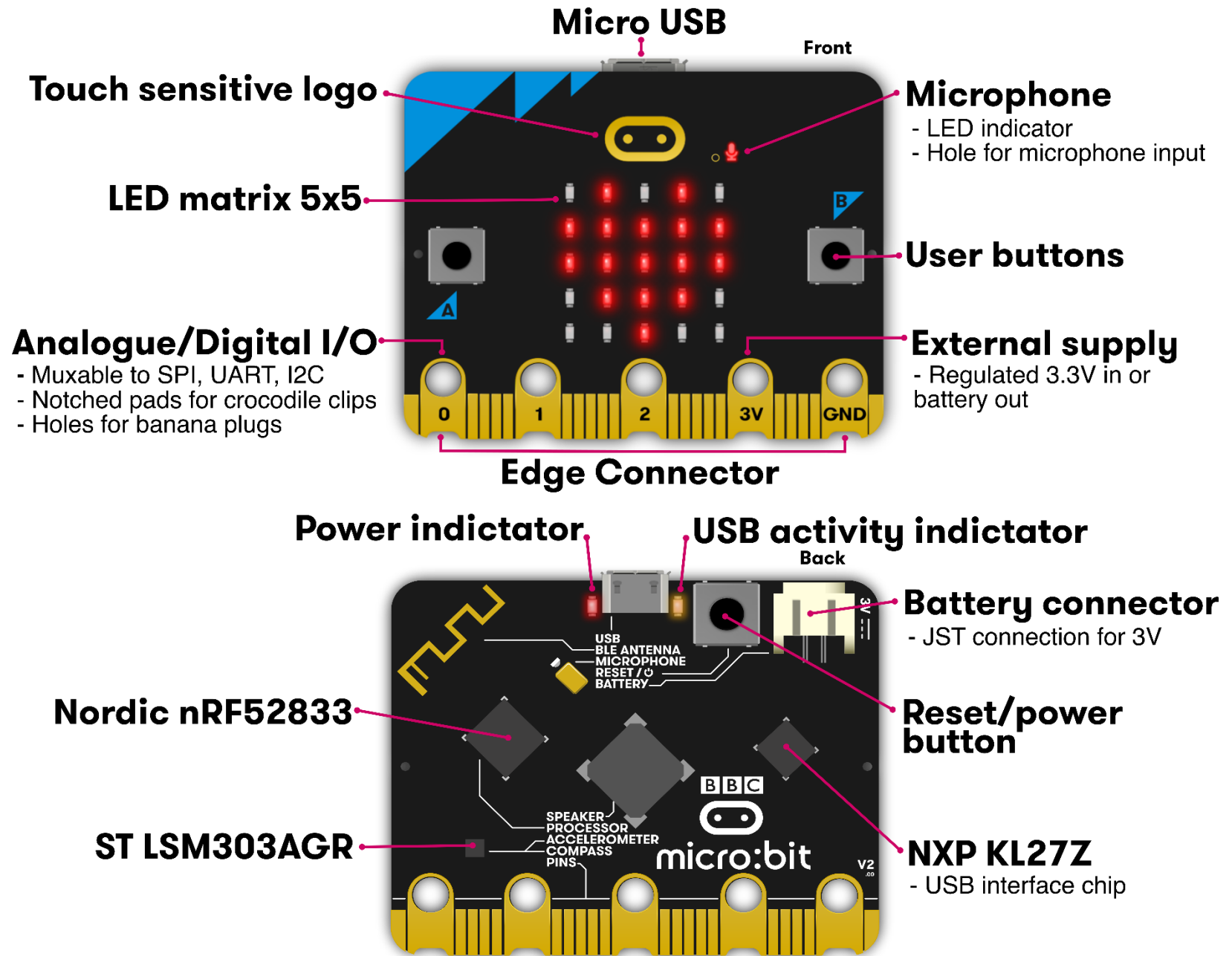
- Explore sensing systems research

Outline

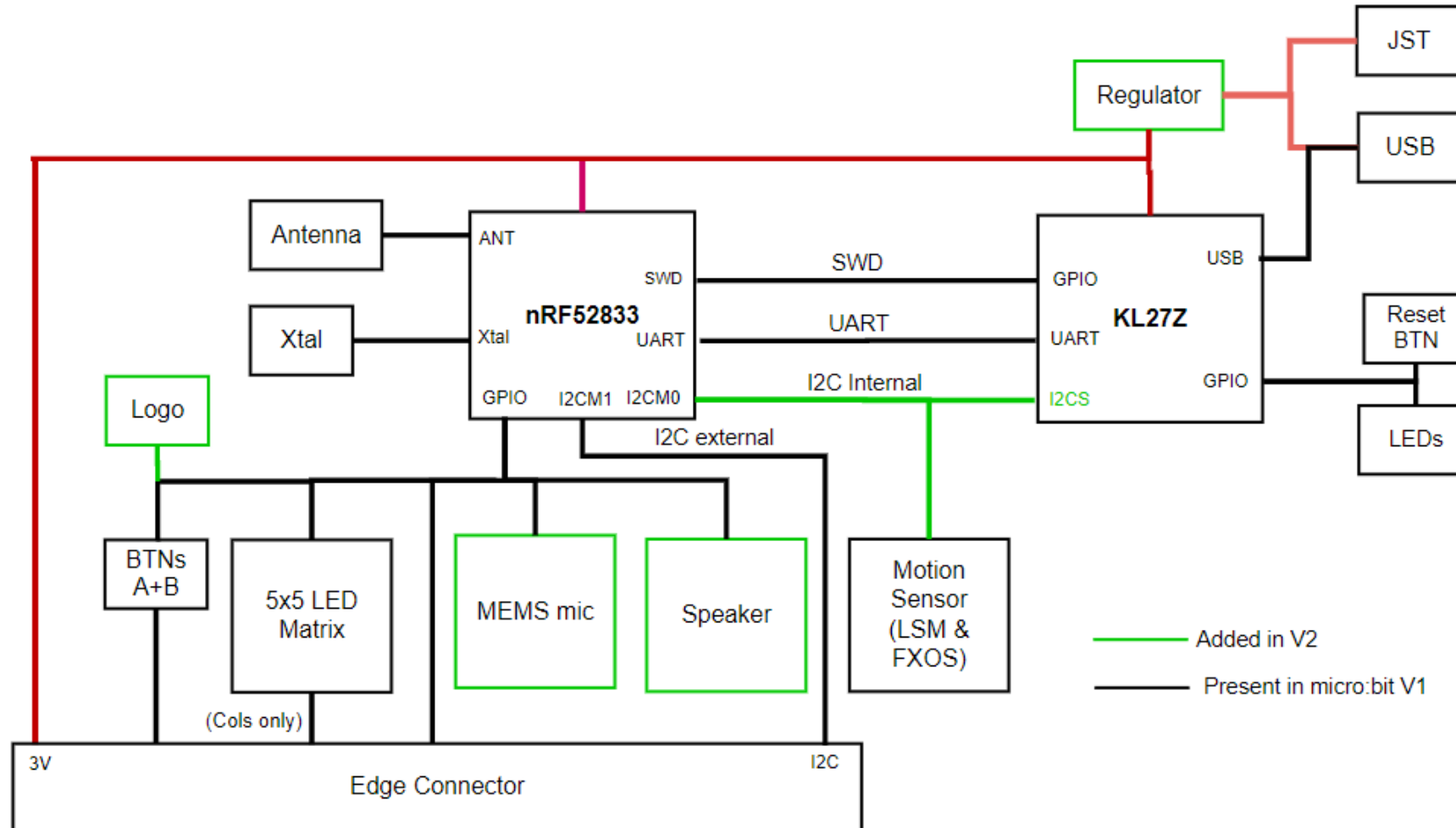- **What haven't we talked about?**
  - **Microbit**
  - nRF52833

- Other hardware systems
  - Other microcontrollers
  - Microprocessors
  - FPGAs

Microbit

- ## Used almost all of this!

- ## Remaining:
  - Batteries
  - Wireless
  - KL27Z I2C

Micro USB

Touch sensitive logo

Microphone
- LED indicator
- Hole for microphone input

LED matrix 5x5

User buttons

Analogue/Digital I/O
- Muxable to SPI, UART, I2C
- Notched pads for crocodile clips
- Holes for banana plugs

External supply
- Regulated 3.3V in or battery out

0   1   2   3V   GND

Edge Connector

Power indictator

USB activity indictator

Back

USB
BLE ANTENNA
MICROPHONE
RESET / ⏻
BATTERY

Battery connector
- JST connection for 3V

Nordic nRF52833

Reset/power button

SPEAKER
PROCESSOR
ACCELEROMETER
COMPASS
PINS

BBC

ST LSM303AGR

micro:bit

V2
.00

NXP KL27Z
- USB interface chip

Internal Microbit connections

KL27 I2C Interface

- Device information
  - Version of board and JTAG firmware
  - Power state of board
    - USB, Battery, both
    - Voltage values for battery and VIN
  - USB connection state
  - Disable the power LED!!

- Flash Storage
  - 128 kB of the KL27's Flash is readable/writable over I2C

Outline

- **What haven't we talked about?**
  - Microbit
  - **nRF52833**

- Other hardware systems
  - Other microcontrollers
  - Microprocessors
  - FPGAs

nRF52833 Peripherals

- Tour of the nRF52833 peripherals

- With some details on the ones we haven't talked about
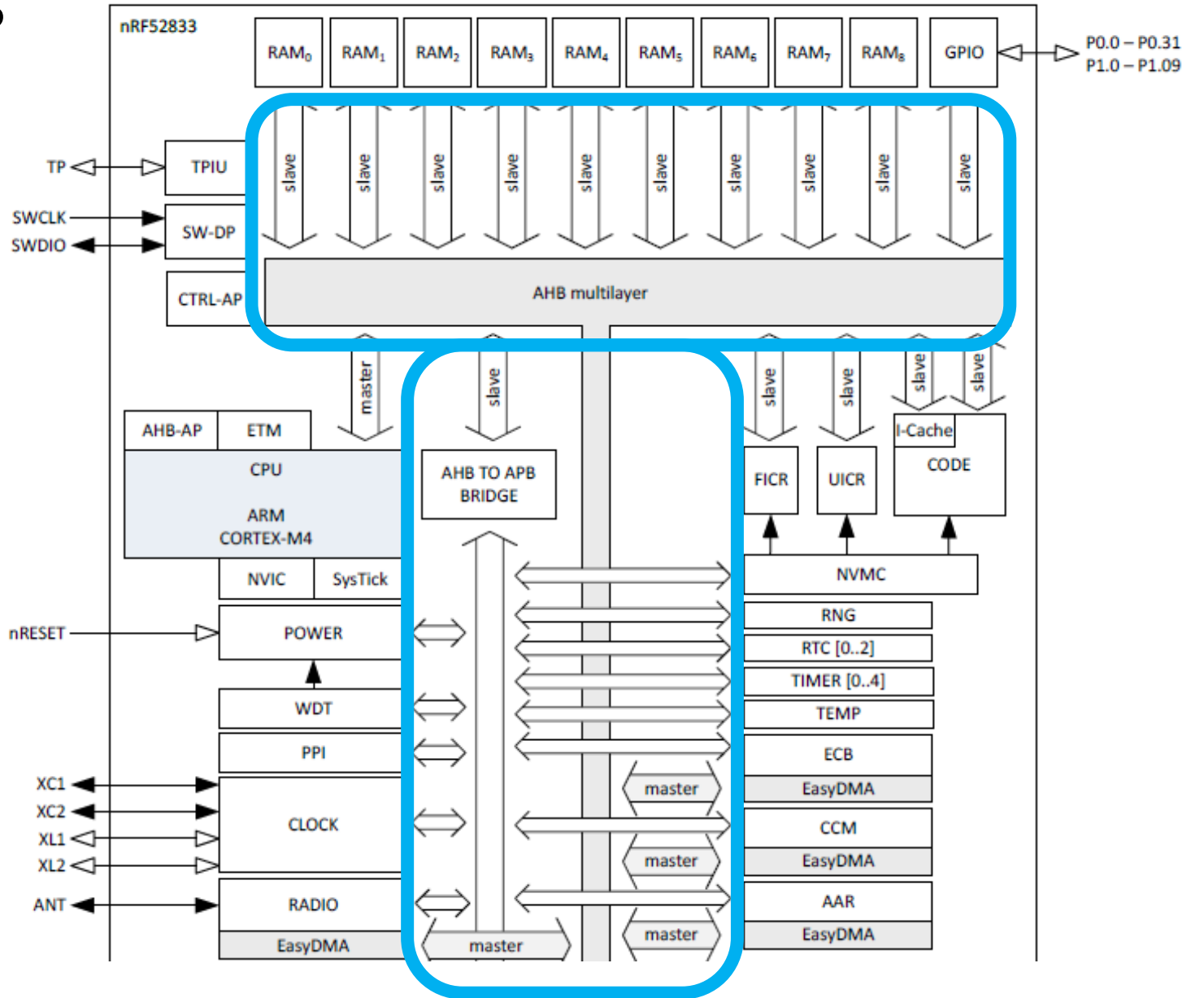 - Wireless
  - Crypto
  - Audio
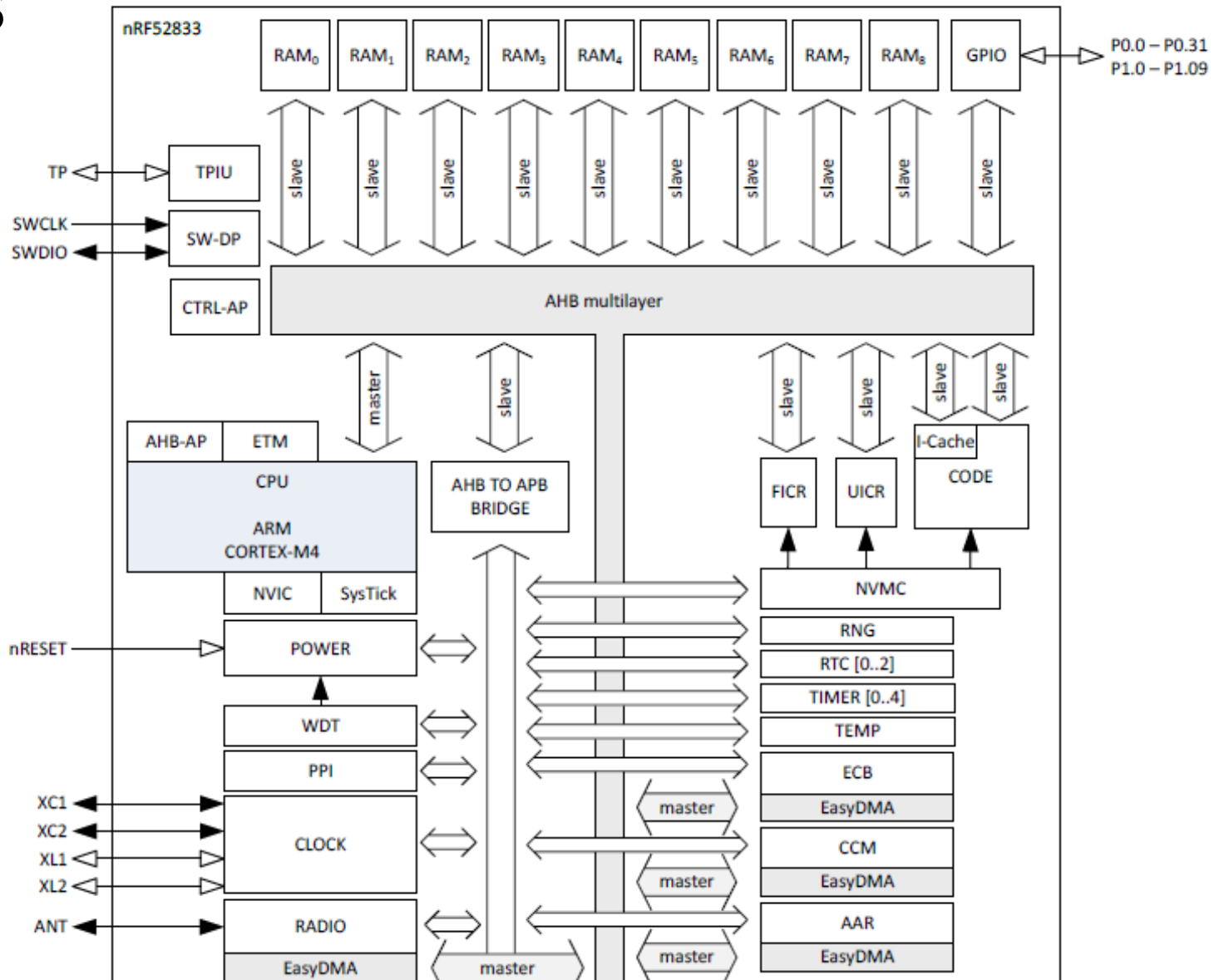
nRF52833 Peripherals

- Cortex-M4F processor

- 32-bit ARM core

- Floating point

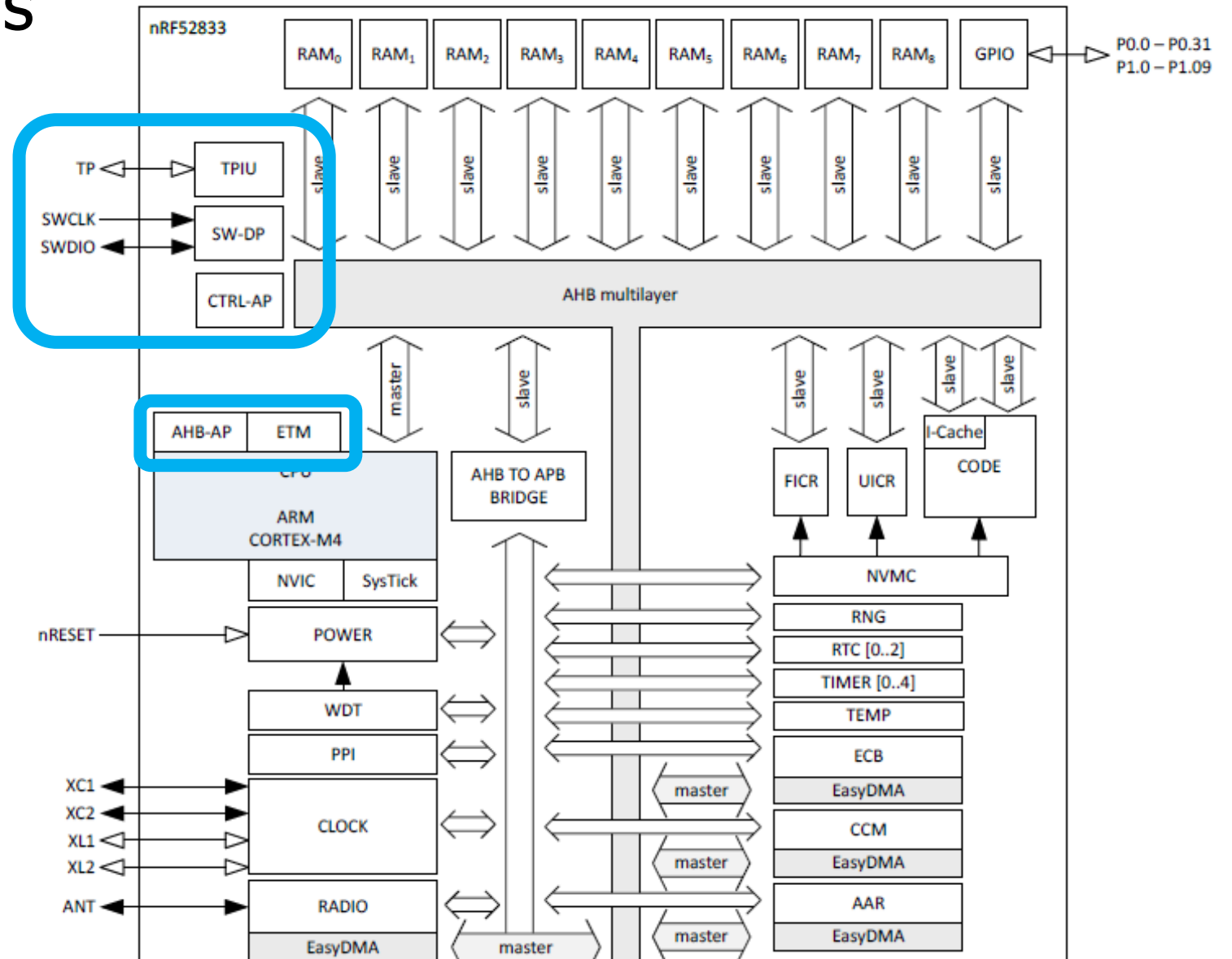- Includes Interrupt control and SysTick (an extra timer)
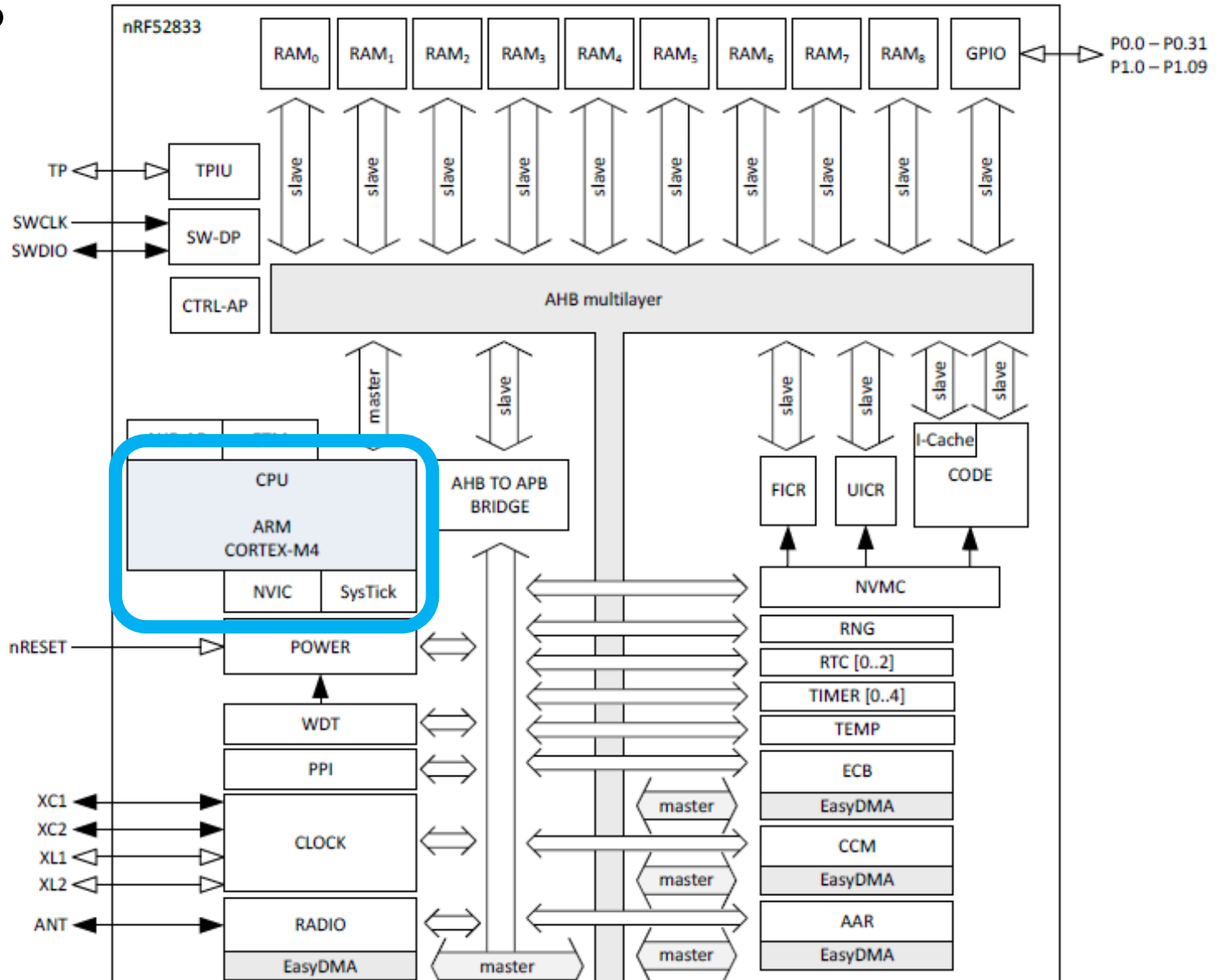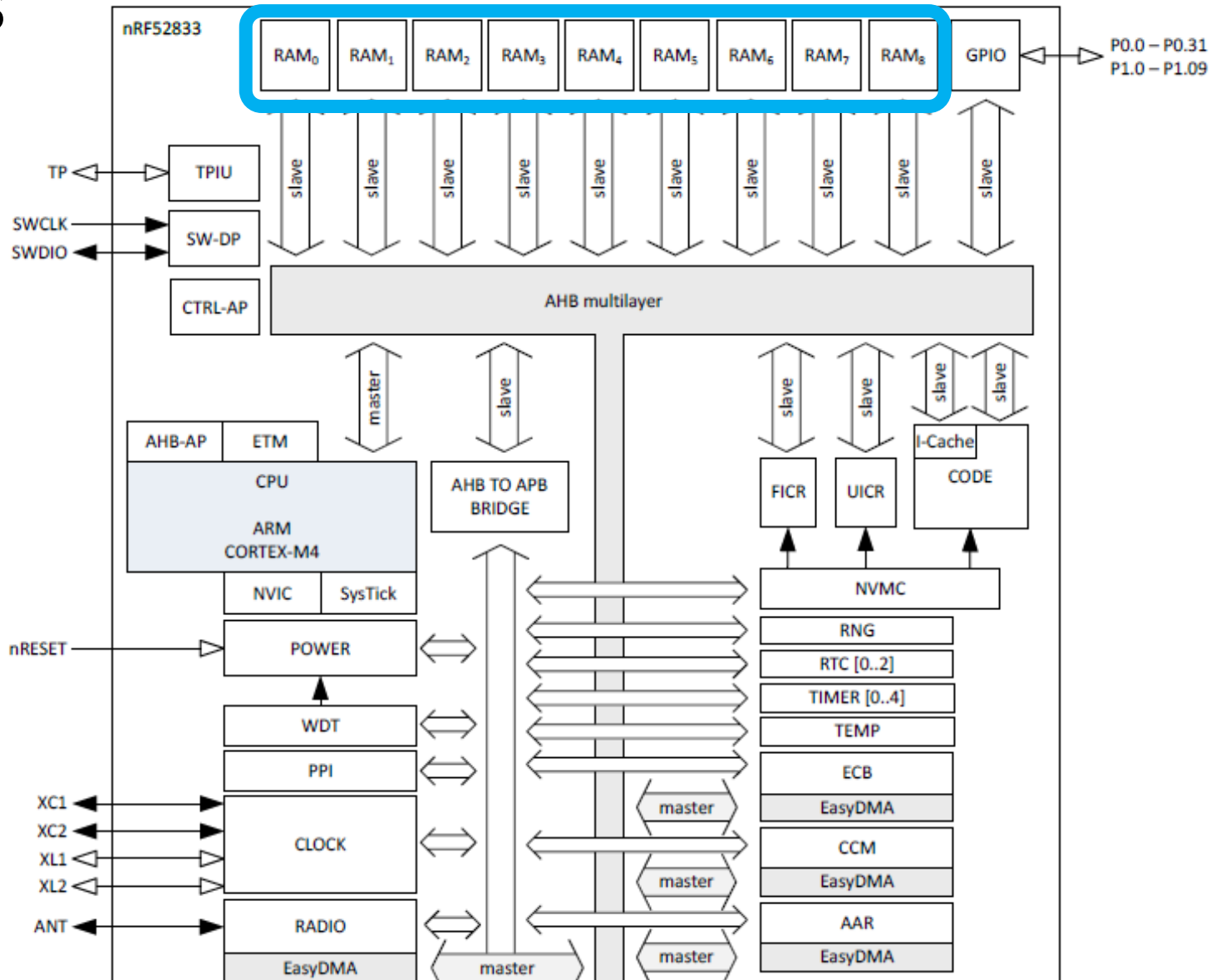
nRF52833 Peripherals

- Memory buses

nRF52833 Peripherals

- JTAG and Debugging

- Allows code updates

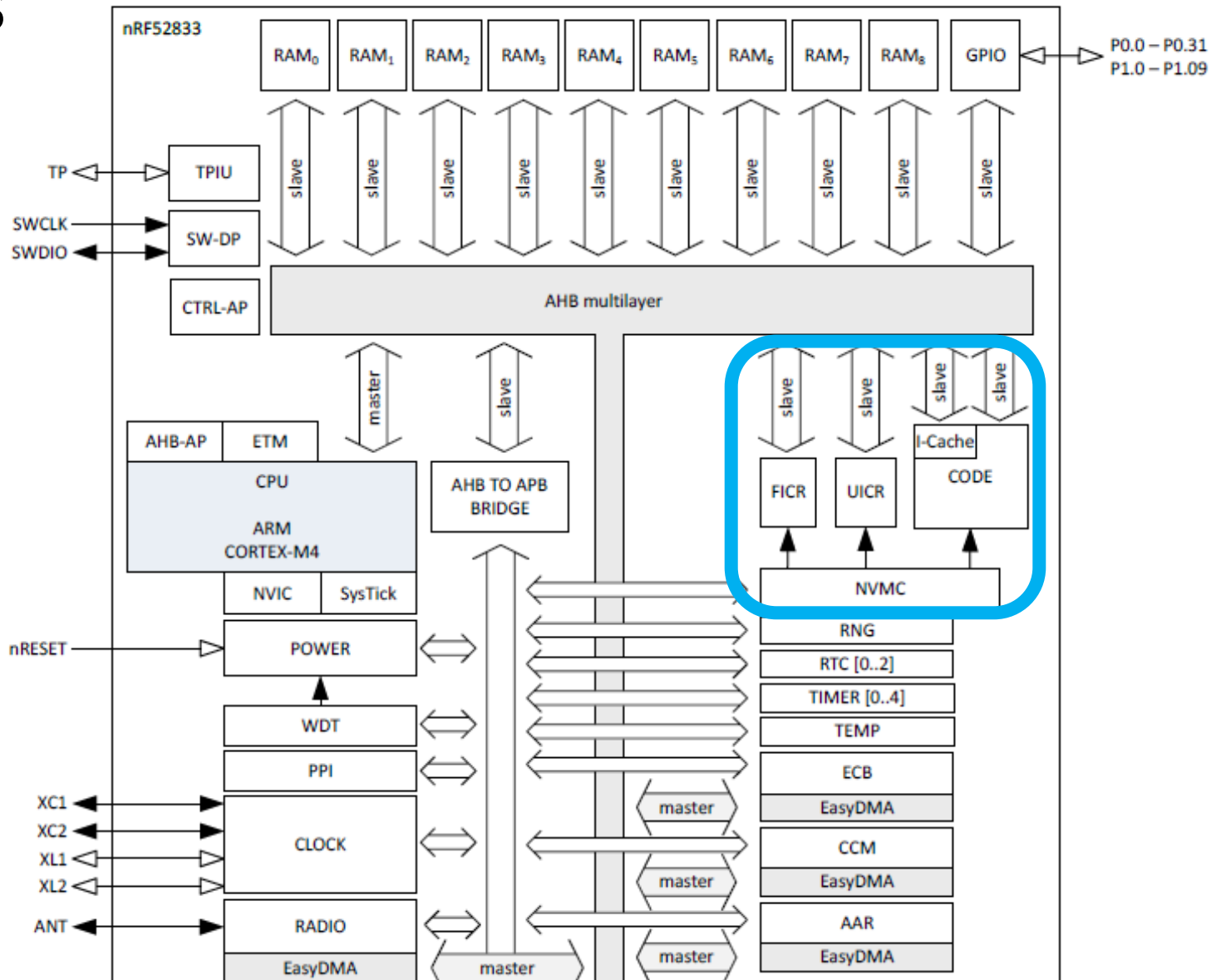- Allows GDB to step through code

nRF52833 Peripherals
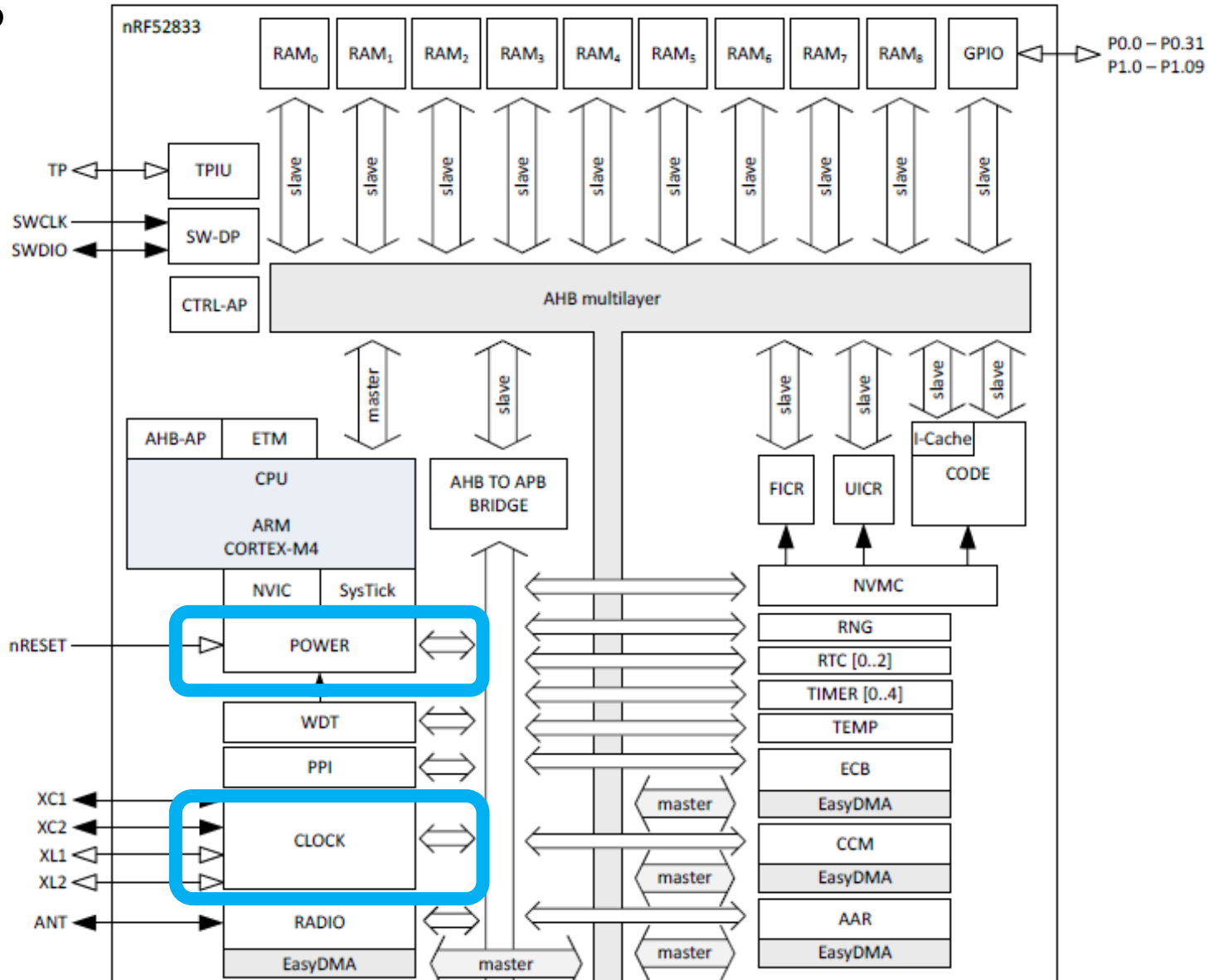
- Volatile memory

- SRAM, 128 kB

nRF52833 Peripherals

- Nonvolatile memory
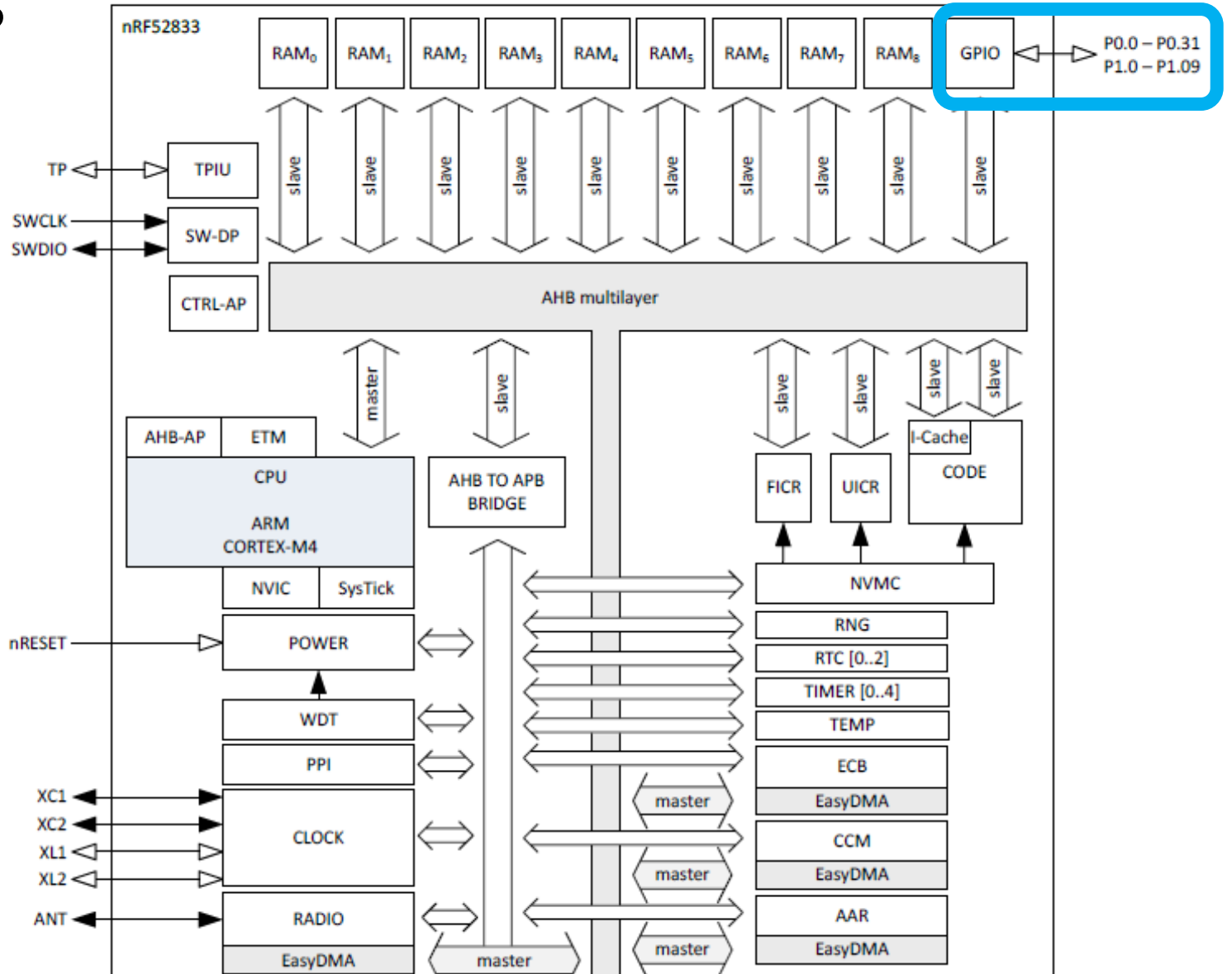
- Flash, 512 kB

- Non-Volatile Memory Controller

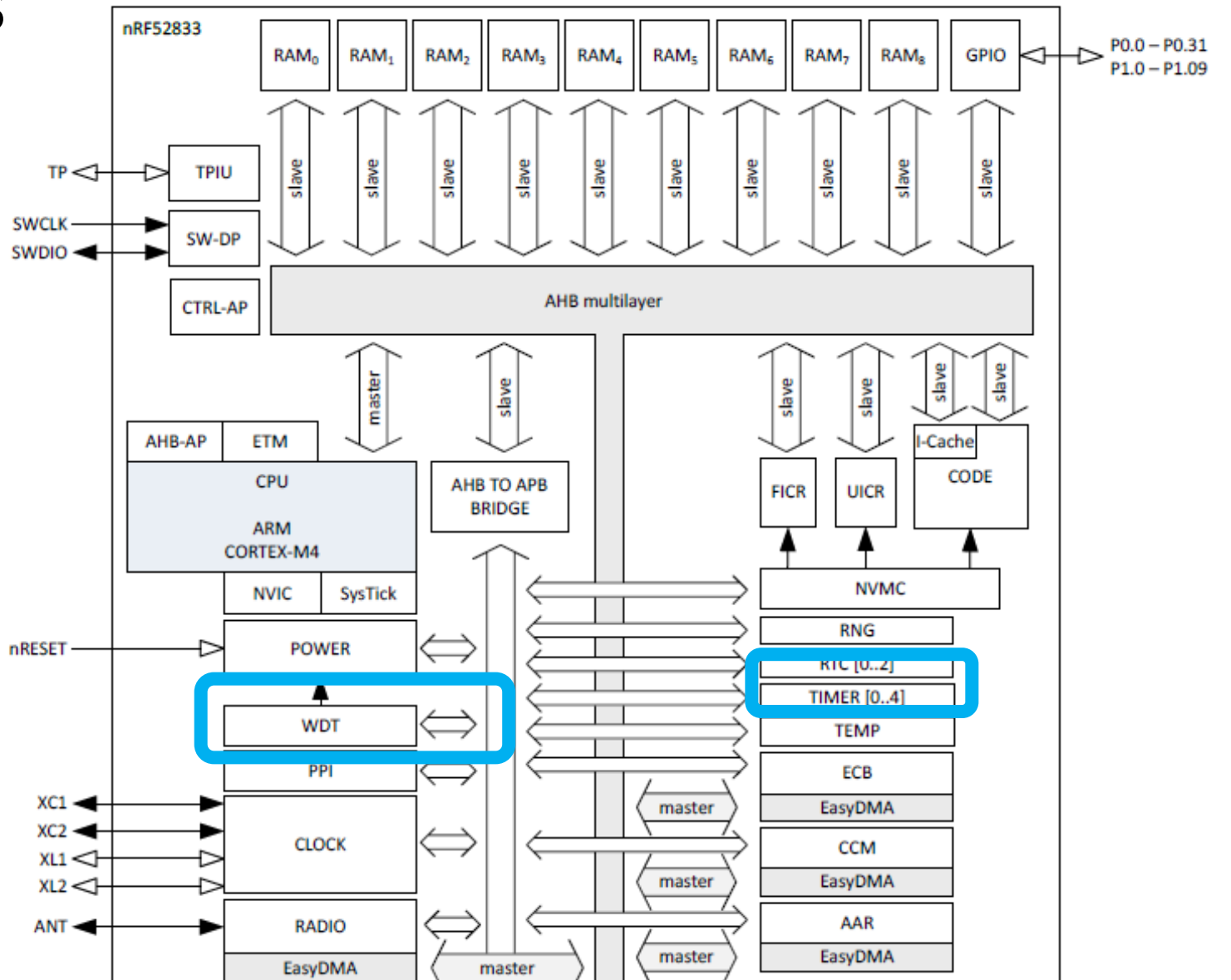# nRF52833 Peripherals

- Power and Clock management

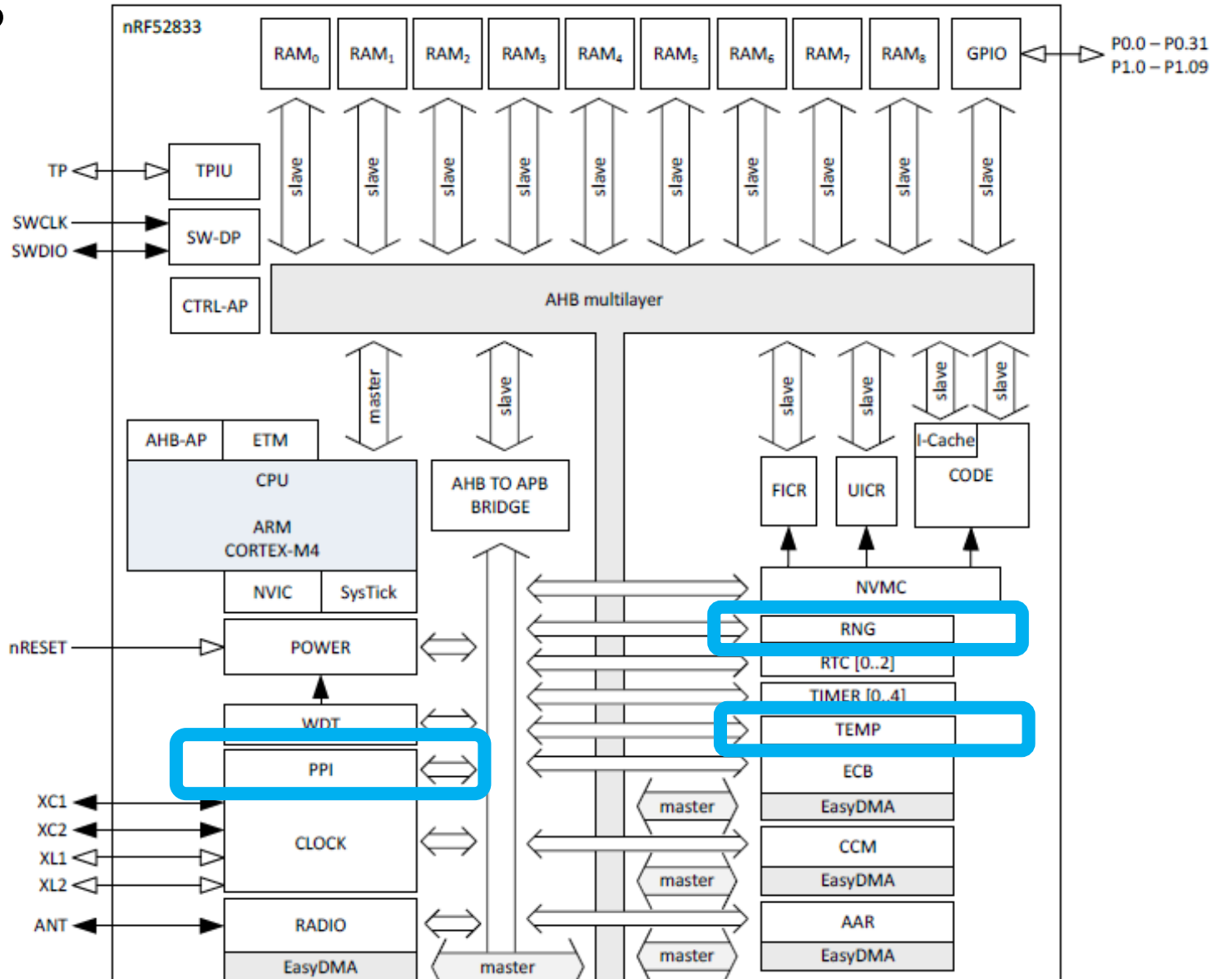# nRF52833 Peripherals

- GPIO pins

nRF52833 Peripherals

- Various timers

- Watchdog Timer

- Real-Time Counter
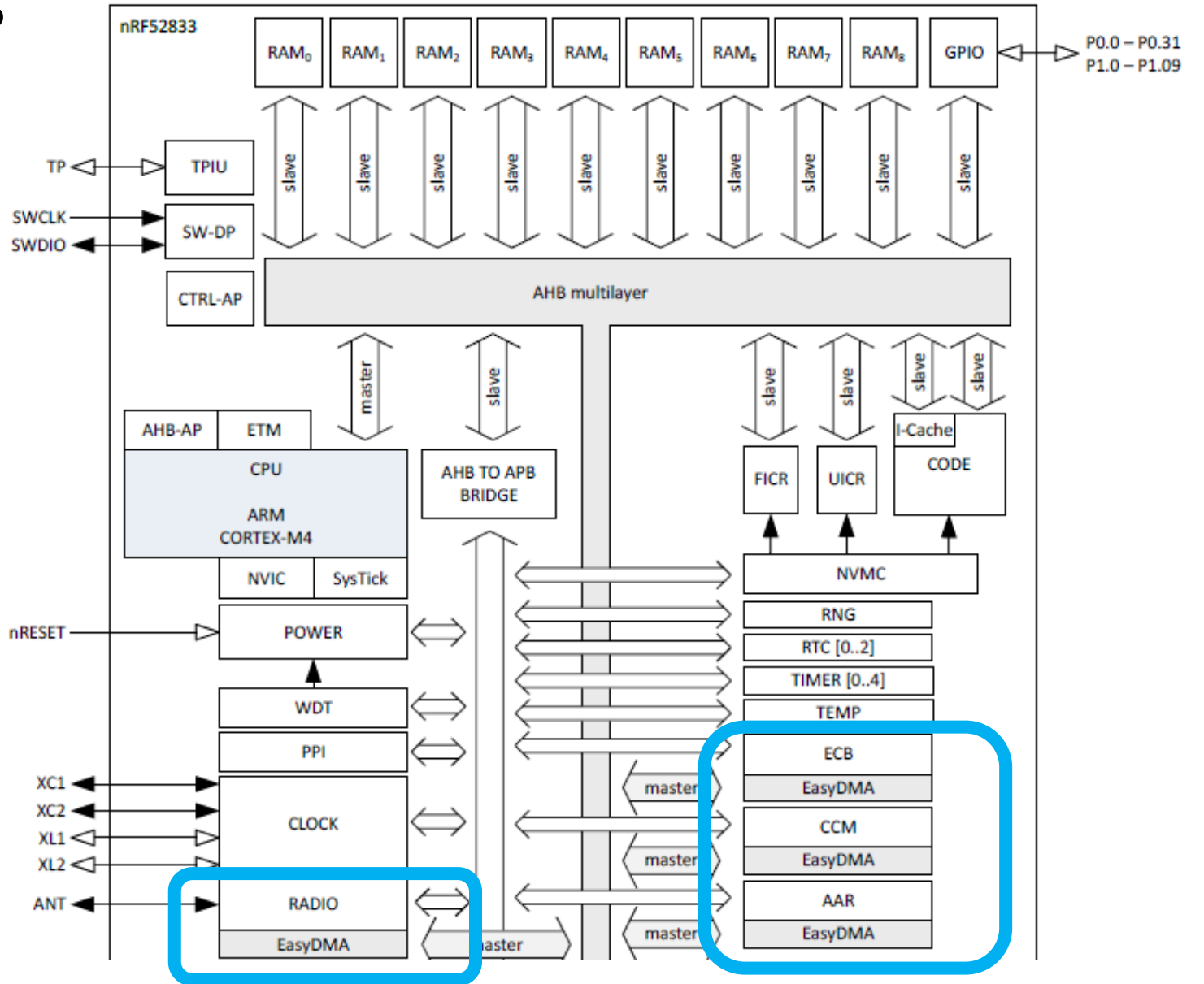
- Timer peripheral

nRF52833 Peripherals

- Programmable Peripheral Interconnect

- Random Number Generator

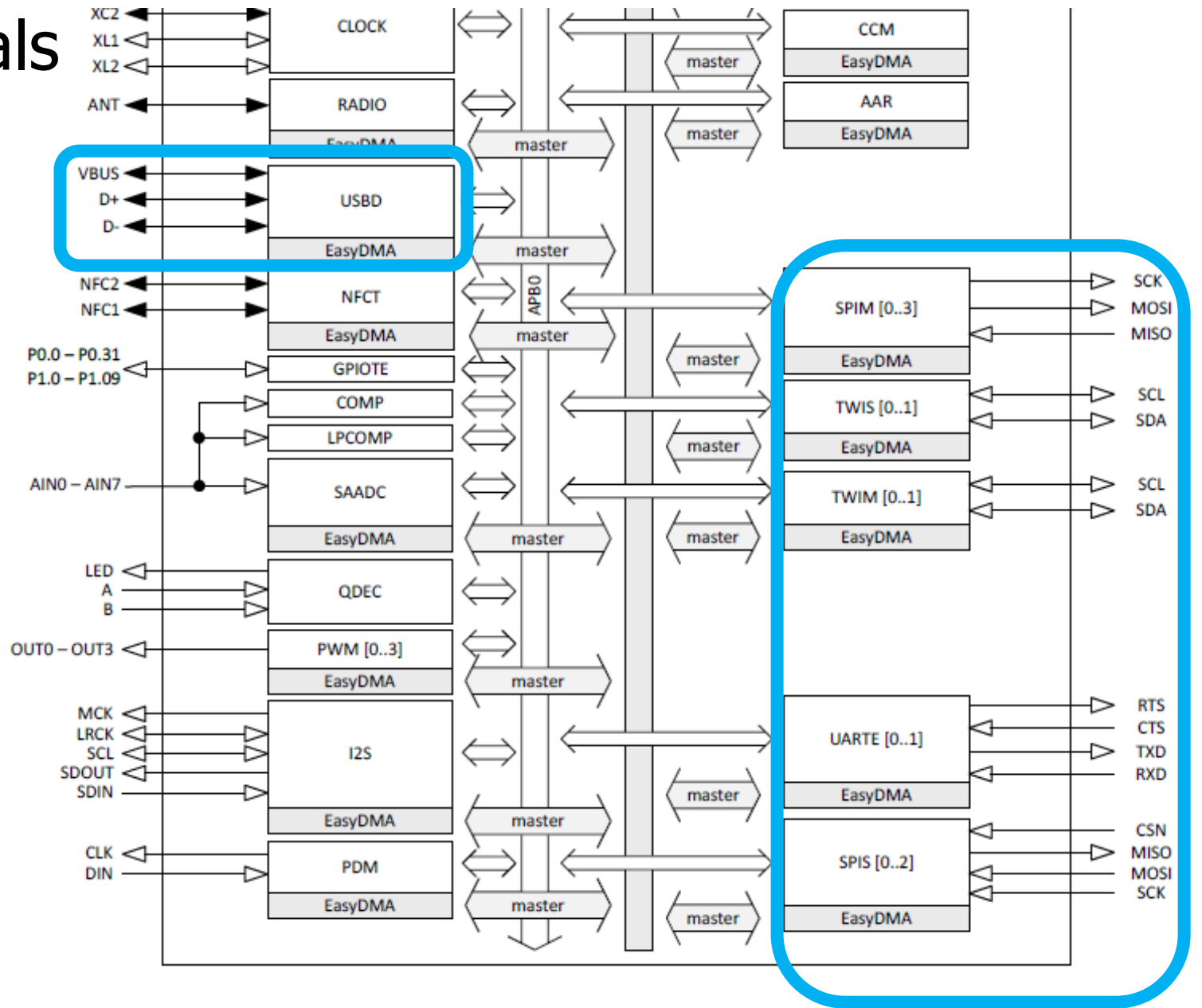- Temperature sensor

nRF52833 Peripherals

- Wireless radio
  - Bluetooth Low Energy
  - 802.15.4
    (Zigbee or Thread)

- Cryptography
  - ECB (AES mode)
  - CCM (AES mode)

  - AAR (Accelerated Address Resolver)
    - For BLE random addresses

nRF52833 Peripherals

- Wired communication protocols

- USB Device

- SPI
  - Controller/Peripheral

- TWI (I2C)
  - Controller/Peripheral

- UART

nRF52833 Peripherals

- Inter-IC Sound (I2S)
  - Wired communication bus explicitly for audio data
  - Unrelated to I2C

- Like a synchronous UART
  - Clock, data in, data out

- Additional signals
  - MCK – synchronization
  - LRCK – left/right channel select
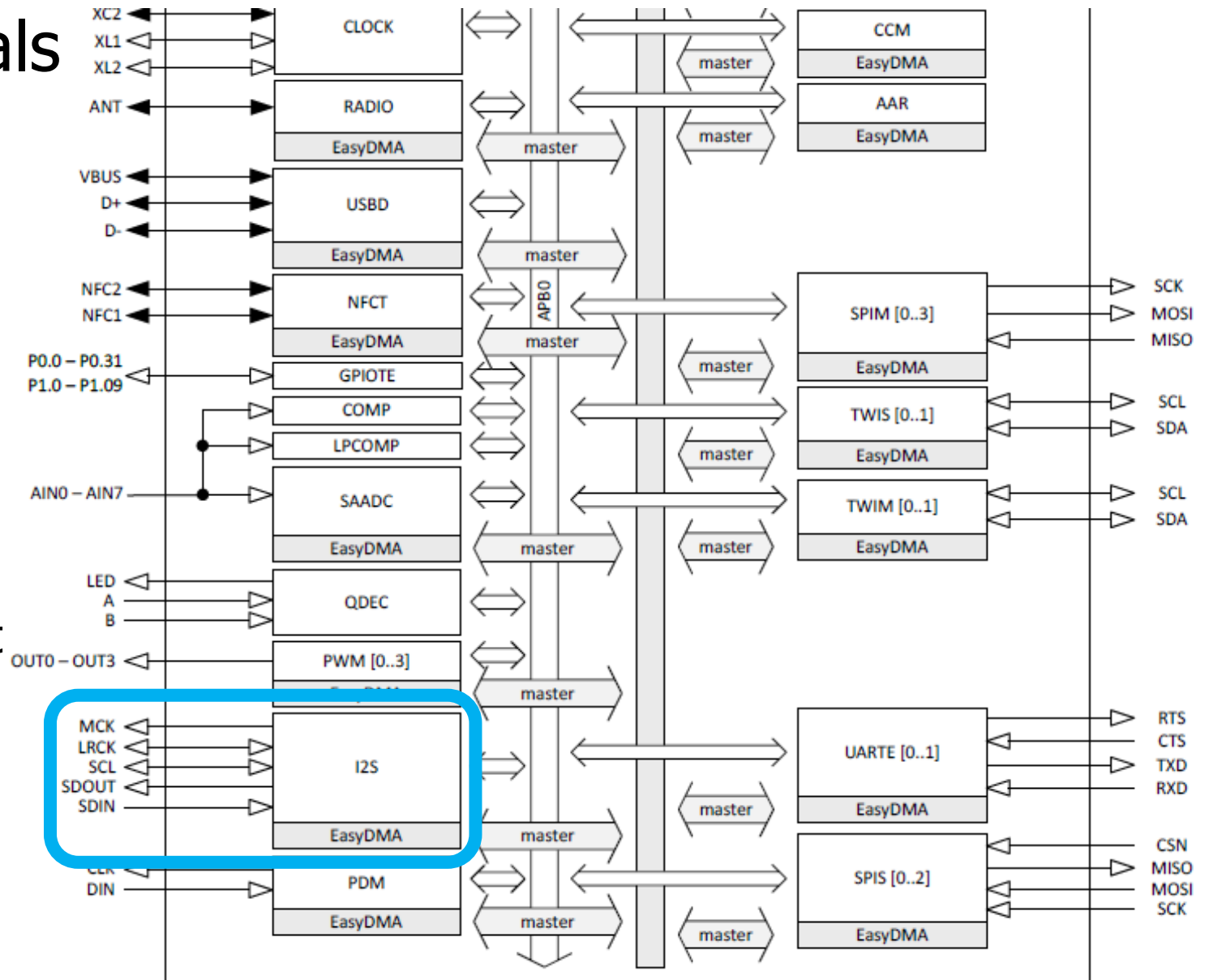
nRF52833 Peripherals

- NFC
  - Near-Field Communication

- Close-range wireless communication protocol

- "Tap-to-pay" systems
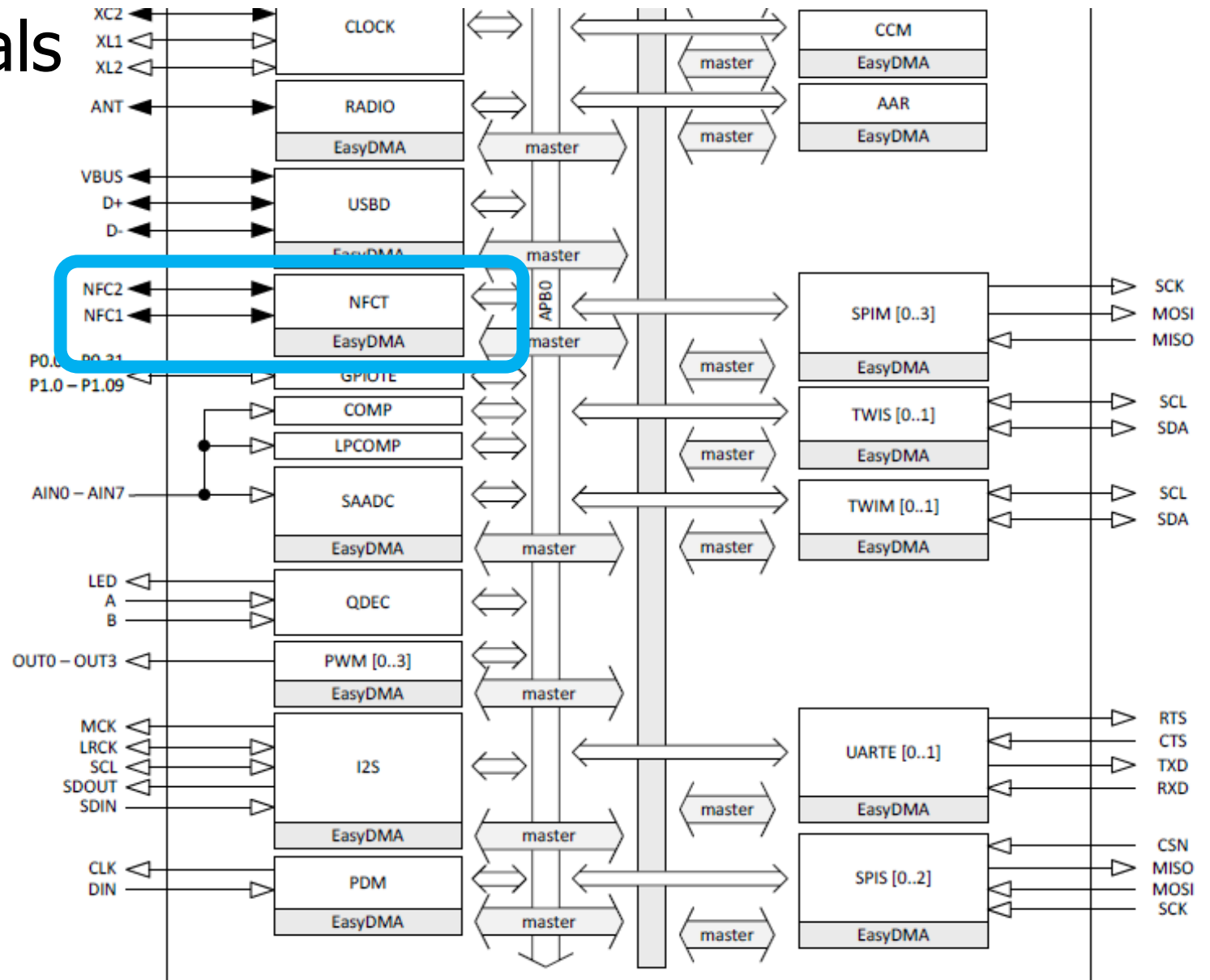
nRF52833 Peripherals

- GPIOTE
  - GPIO interrupts
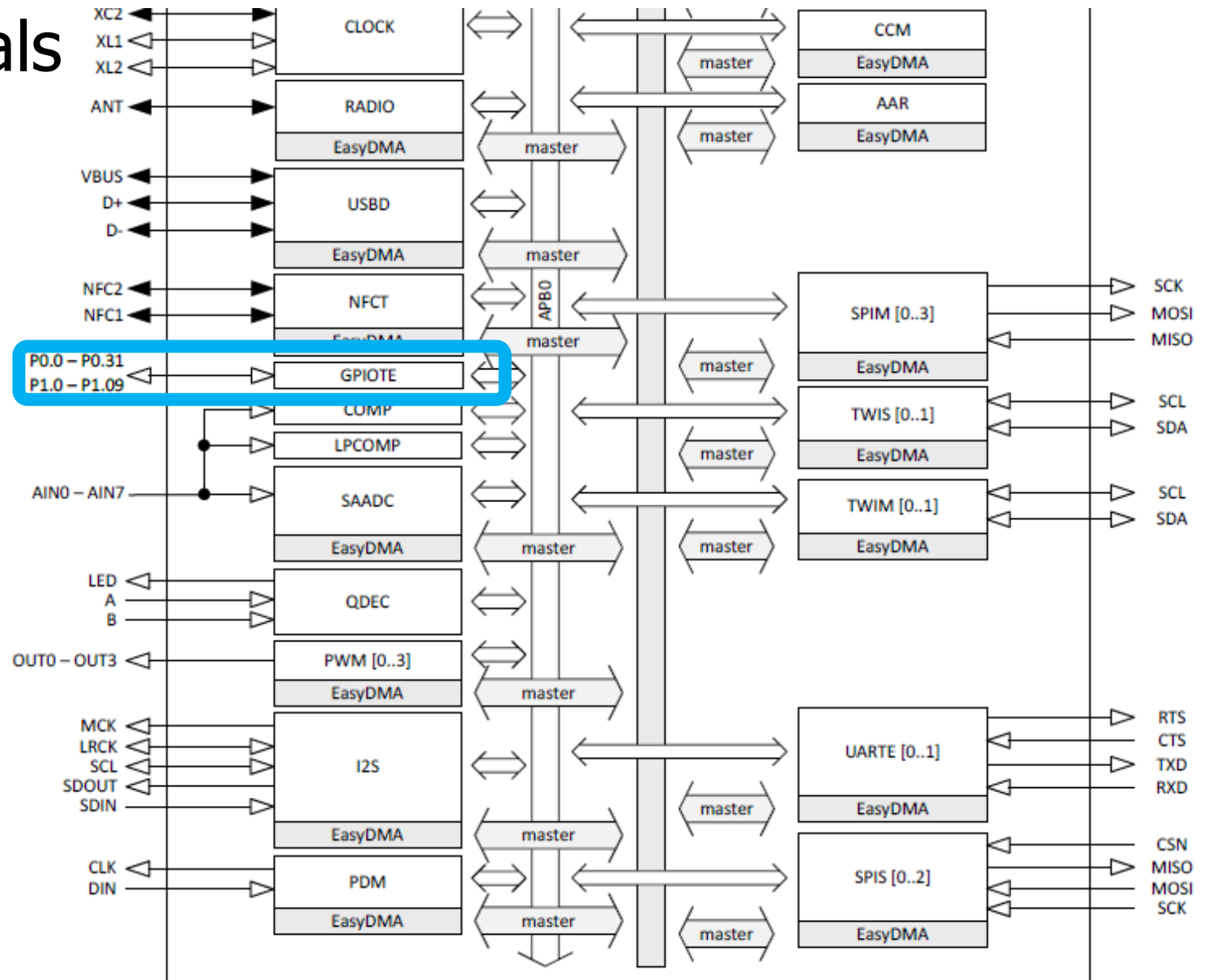
nRF52833 Peripherals

- Analog inputs

- Comparator

- Low-Power Comparator

- Successive Approximation Analog-to-Digital Converter
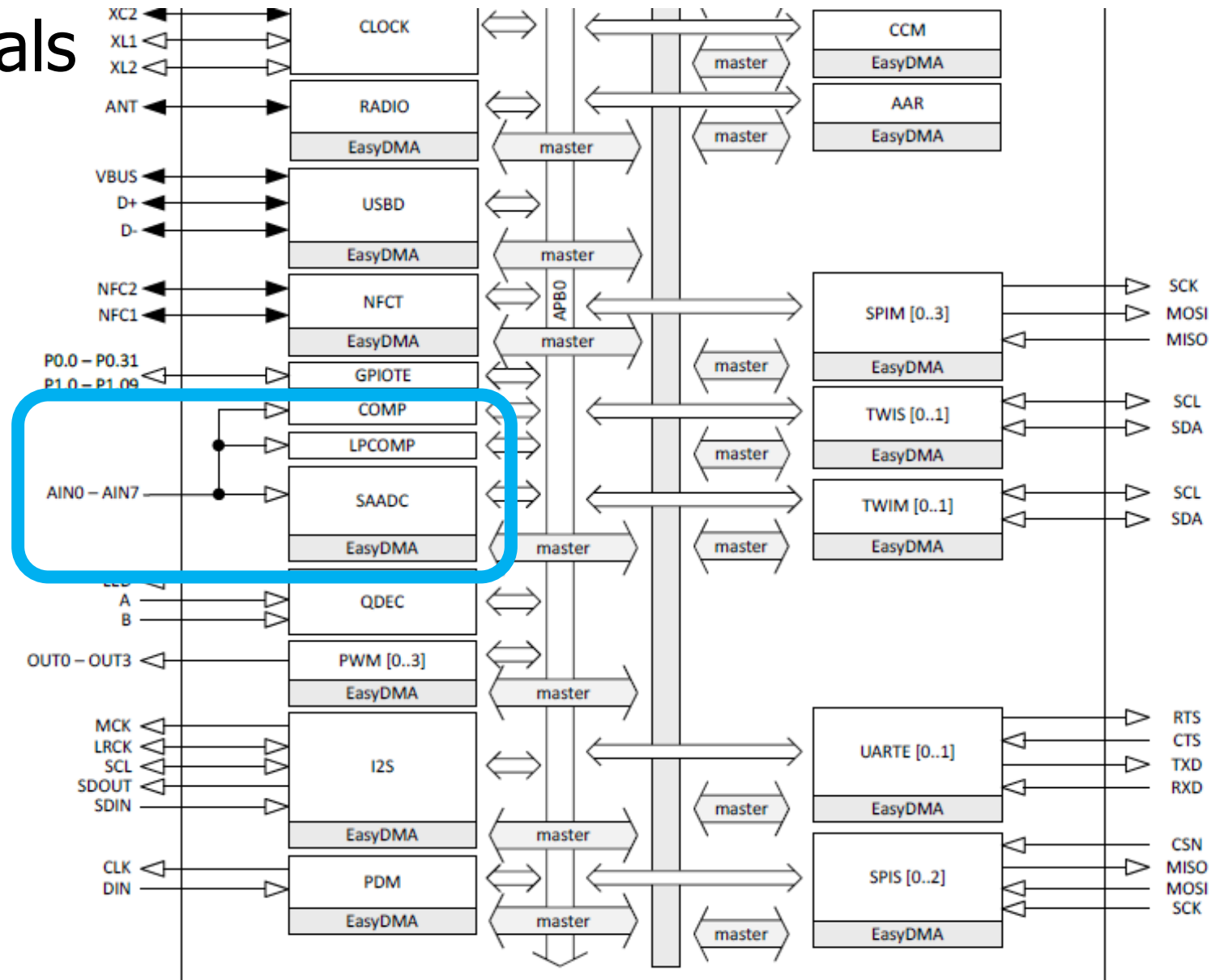
nRF52833 Peripherals

- Quadrature Decoder peripheral

- Detects rotation speeds and direction
  - Usually for motors
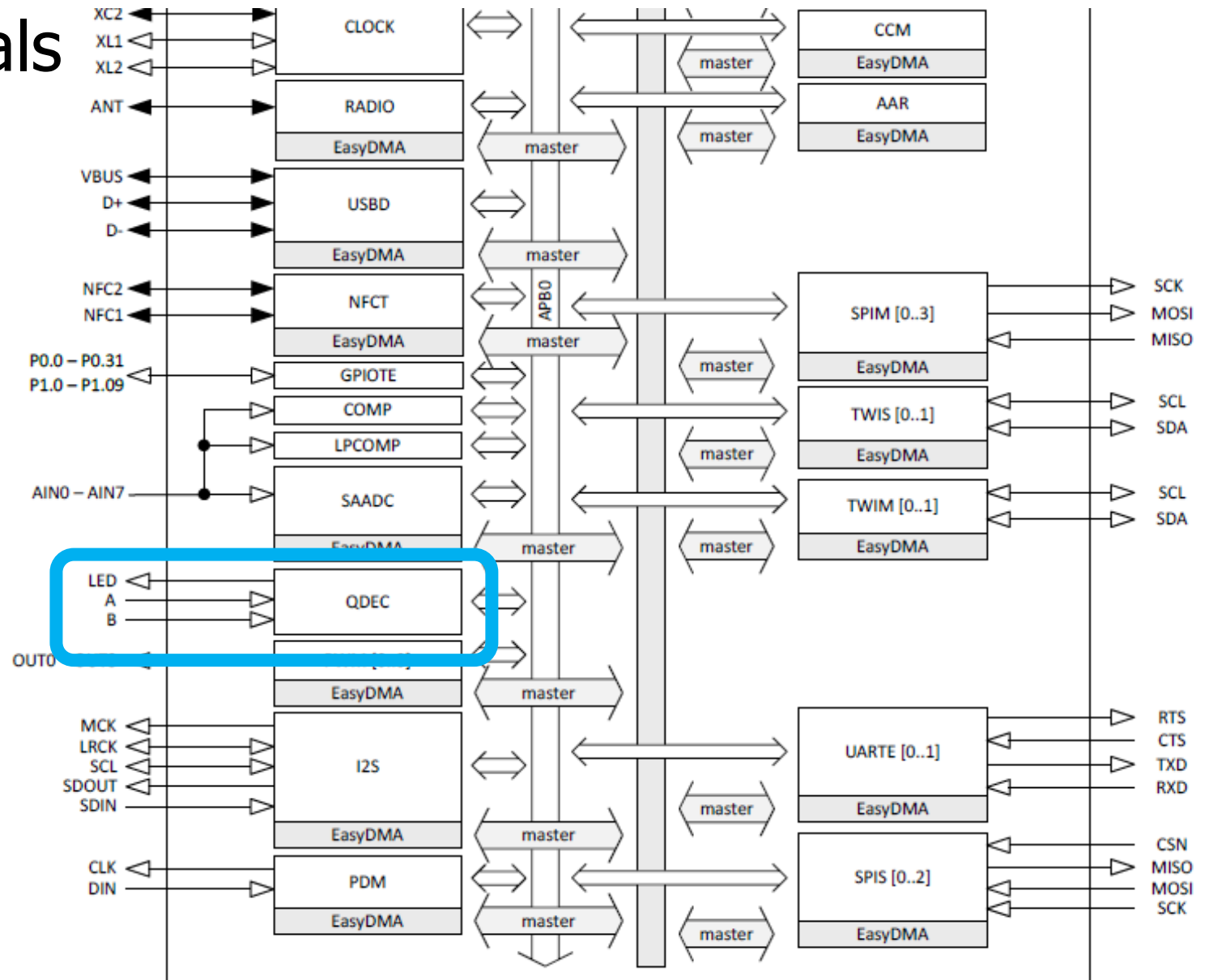
Quadrature Encoding
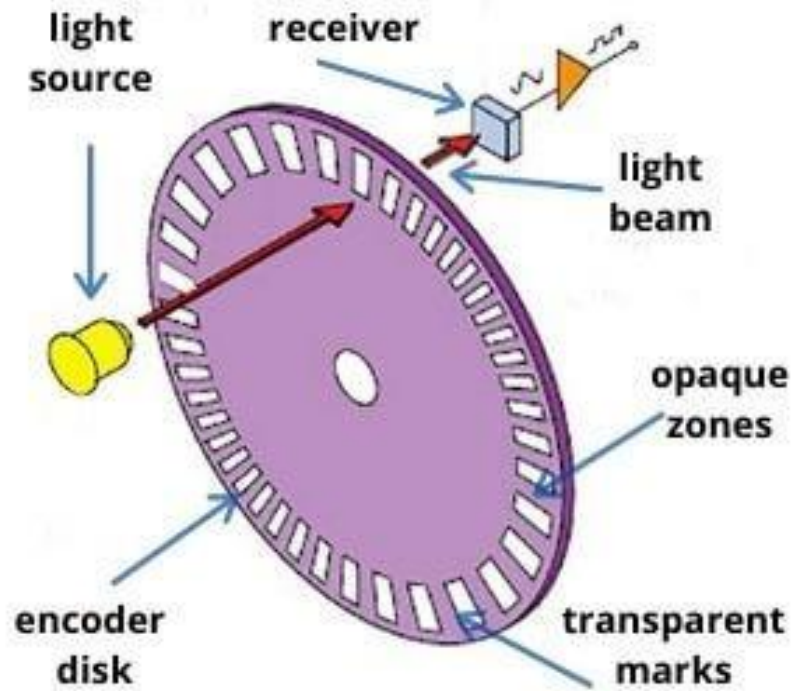
Optical encoder working principle
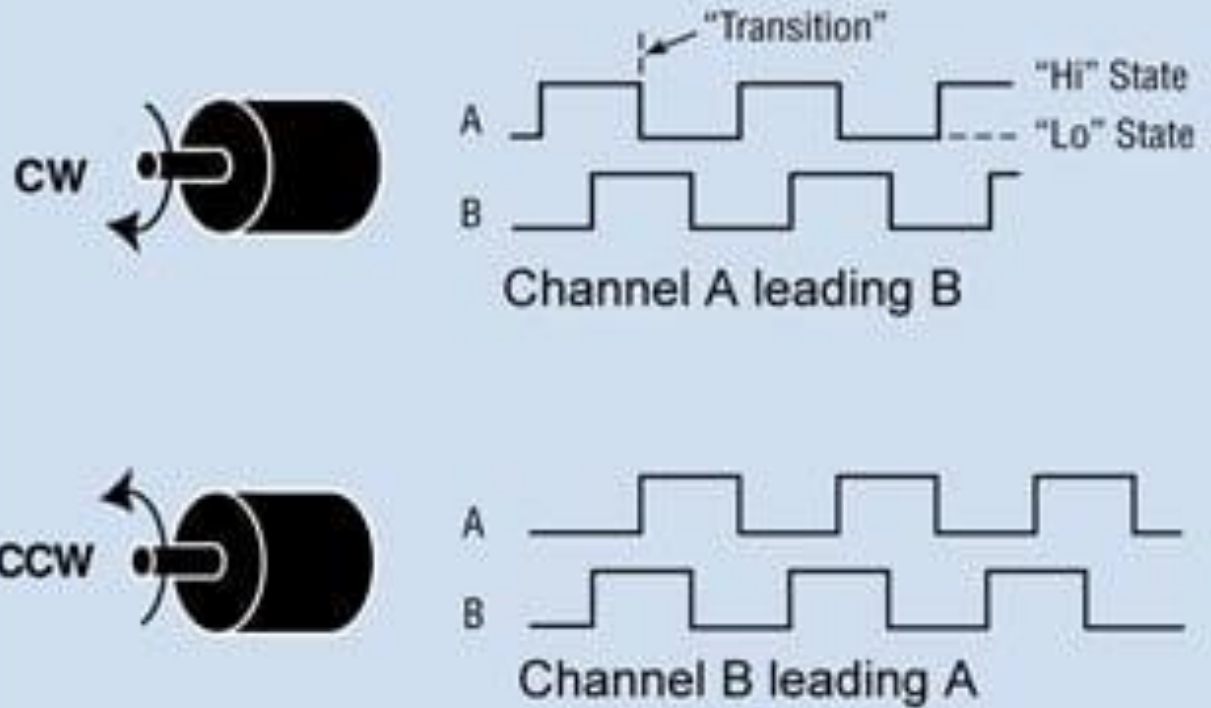
nRF52833 Peripherals

- Pulse Width Modulation

- Pulse Density Modulation
  - Similar idea to PWM
  - Input-only peripheral
  - Targets microphones
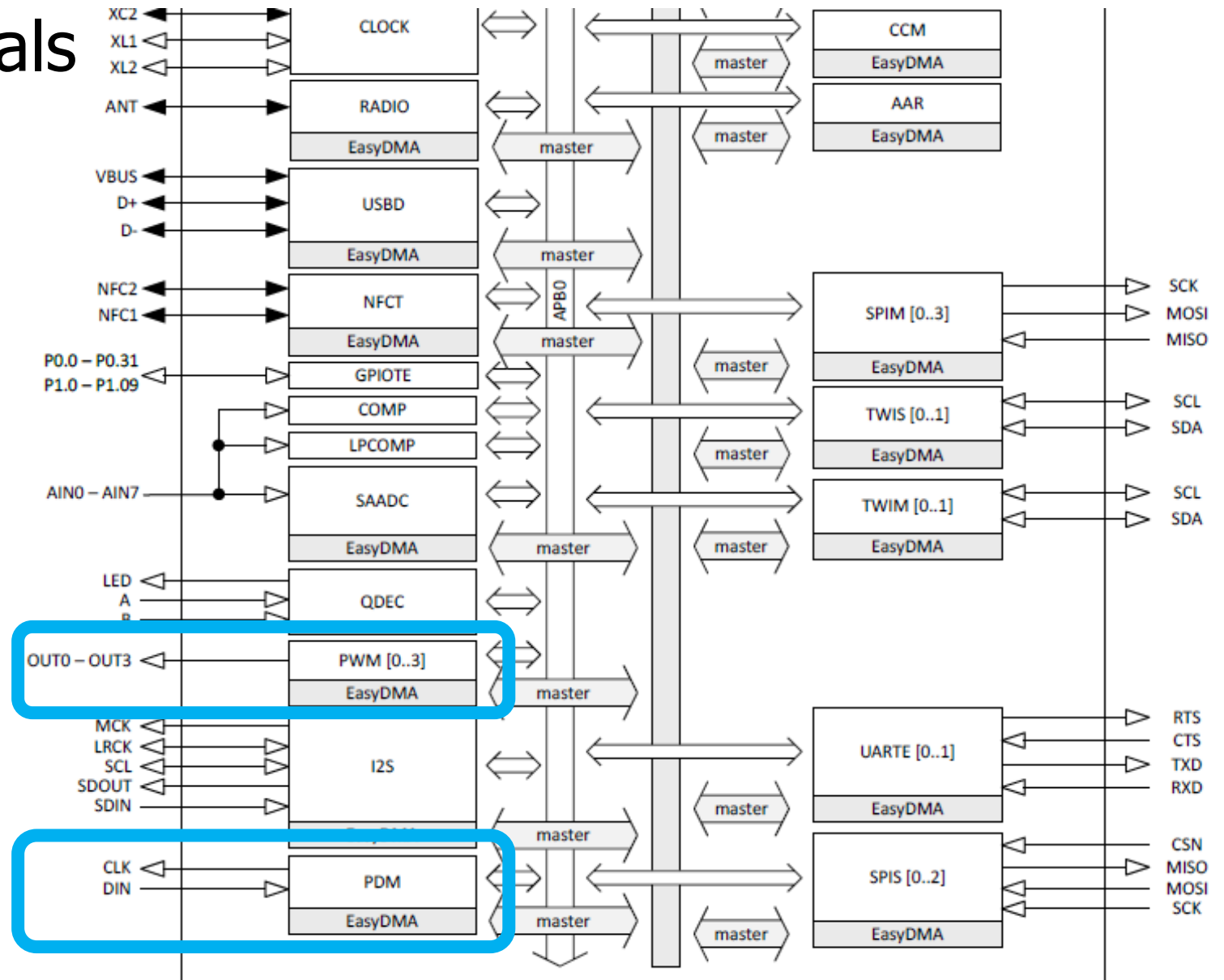
nRF52833 is complete

- That's just about everything!

- First 550 out of 600 pages of nRF52833 datasheet
  - Remaining 50 are hardware details
    - Pinout for different packages
    - Recommended circuit layout
    - Soldering details

Break + Open Question

- What **doesn't** the nRF52833 have a peripheral for?
  (i.e., what could you imagine a peripheral for that it doesn't have)

Outline

- What haven't we talked about?
  - Microbit
  - nRF52833

- **Other hardware systems**
  - **Other microcontrollers**
  - Microprocessors
  - FPGAs

Microcontroller knowledge is transferrable

- Your knowledge of microcontrollers applies to almost all of them
  - They have similar peripherals that work in similar ways

- The exact names and configurations will definitely be different
  - But the fundamentals stay the same

- Let's prove this with a quick view of two microcontrollers
  1. Atmel SAM4L
  2. Texas Instruments MSP430

Atmel SAM4L Microcontroller

- ARM Cortex M4F (same processor!)
- Lots of very configurable peripherals
  - USART, SPI, TWI, I2S, DAC, ADC, Timer, …
  - Kind of a "do-everything" microcontroller
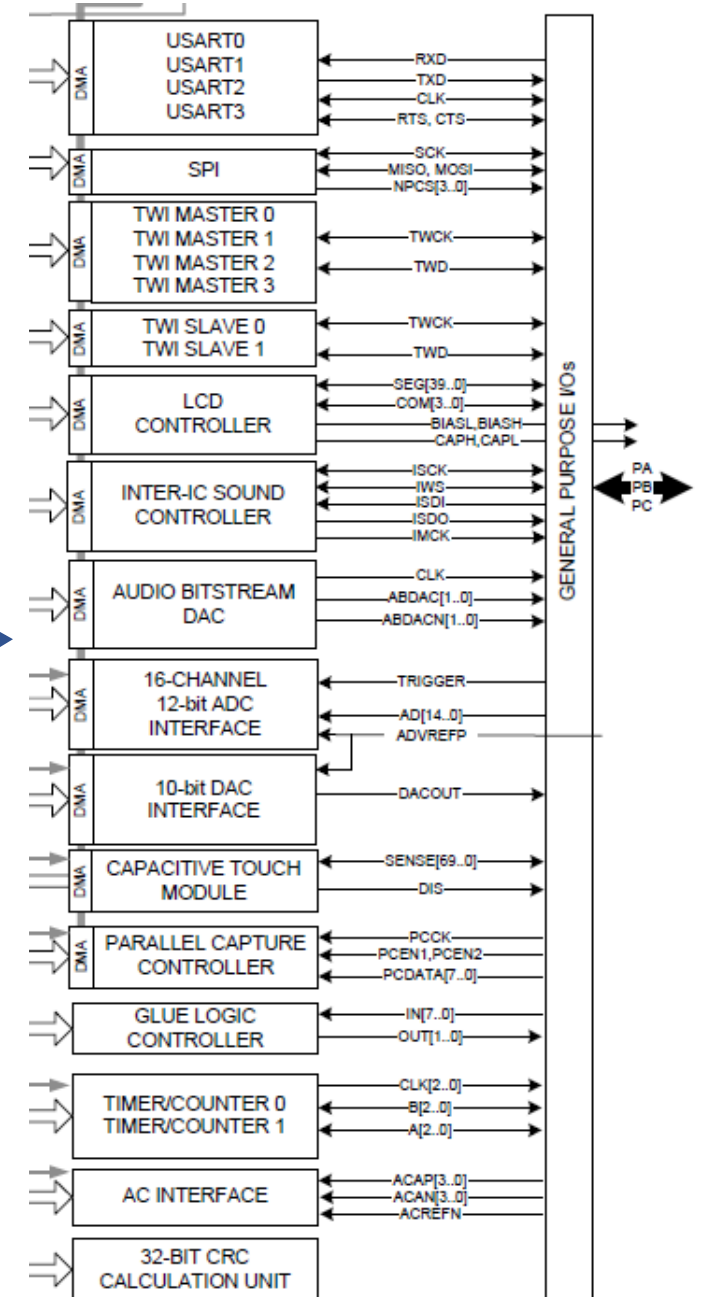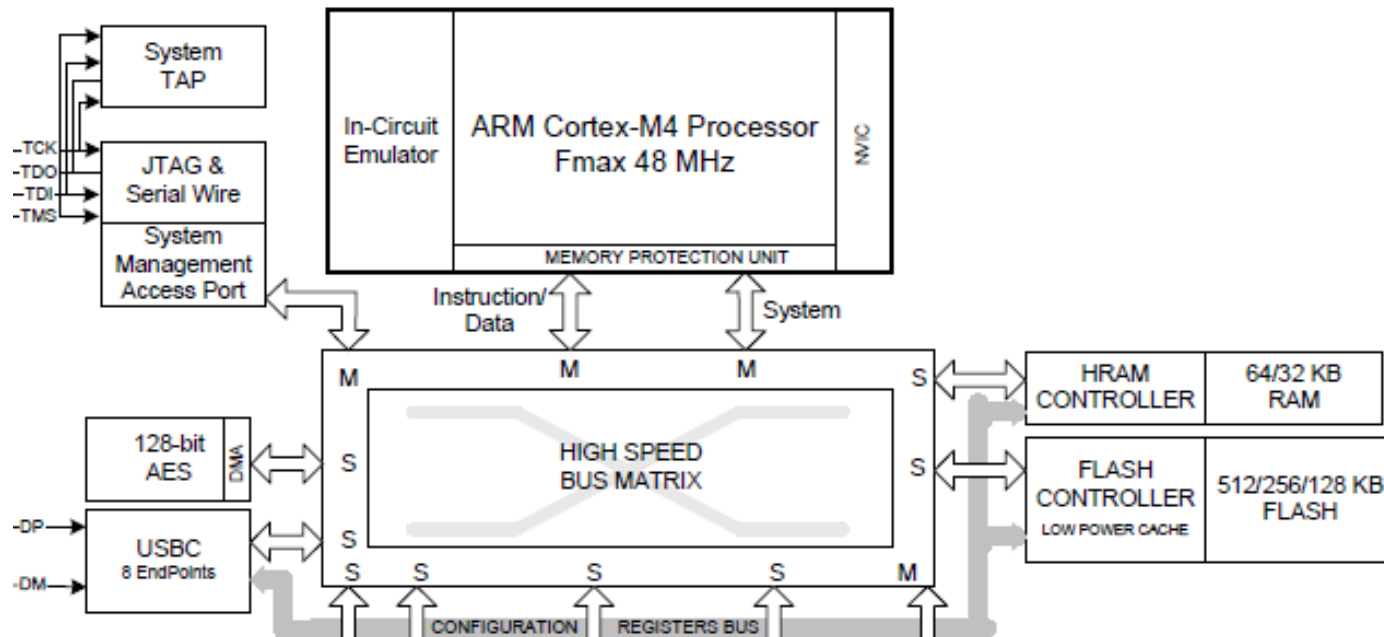
SAM4L GPIO

- GPIO register map (heavily abbreviated)

Table 23-2. GPIO Register Memory Map

Offset	Register	Function	Register Name	Access	Reset	Config. Protection	Access Protection
0x000	GPIO Enable Register	Read/Write	GPER	Read/Write	-[1]	Y	N
0x040	Output Driver Enable Register	Read/Write	ODER	Read/Write	-[1]	Y	N
0x050	Output Value Register	Read/Write	OVR	Read/Write	-[1]	N	N
0x060	Pin Value Register	Read	PVR	Read-only	Depending on pin states	N	N
0x070	Pull-up Enable Register	Read/Write	PUER	Read/Write	-[1]	Y	N
0x080	Pull-down Enable Register	Read/Write	PDER	Read/Write	(1)	Y	N
0x090	Interrupt Enable Register	Read/Write	IER	Read/Write	-[1]	N	N

Setting or clearing individual bits

- Actually, each register appears four times in a row
  - Read/write version
  - Set version
  - Clear version
  - Toggle version

Table 23-2. GPIO Register Memory Map

Offset	Register	Function	Register Name	Access	Reset	Config. Protection	Access Protection
0x000	GPIO Enable Register	Read/Write	GPER	Read/Write	-[1]	Y	N
0x004	GPIO Enable Register	Set	GPERS	Write-only		Y	N
0x008	GPIO Enable Register	Clear	GPERC	Write-only		Y	N
0x00C	GPIO Enable Register	Toggle	GPERT	Write-only		Y	N

SAM4L Timers

- Three 16-bit timers
  - Each of which can have multiple inputs clock signals with prescaler values

  - Timers can be chained together to make up to a 48-bit timer

- One register for reading counter

- Three registers for "compare interrupts"
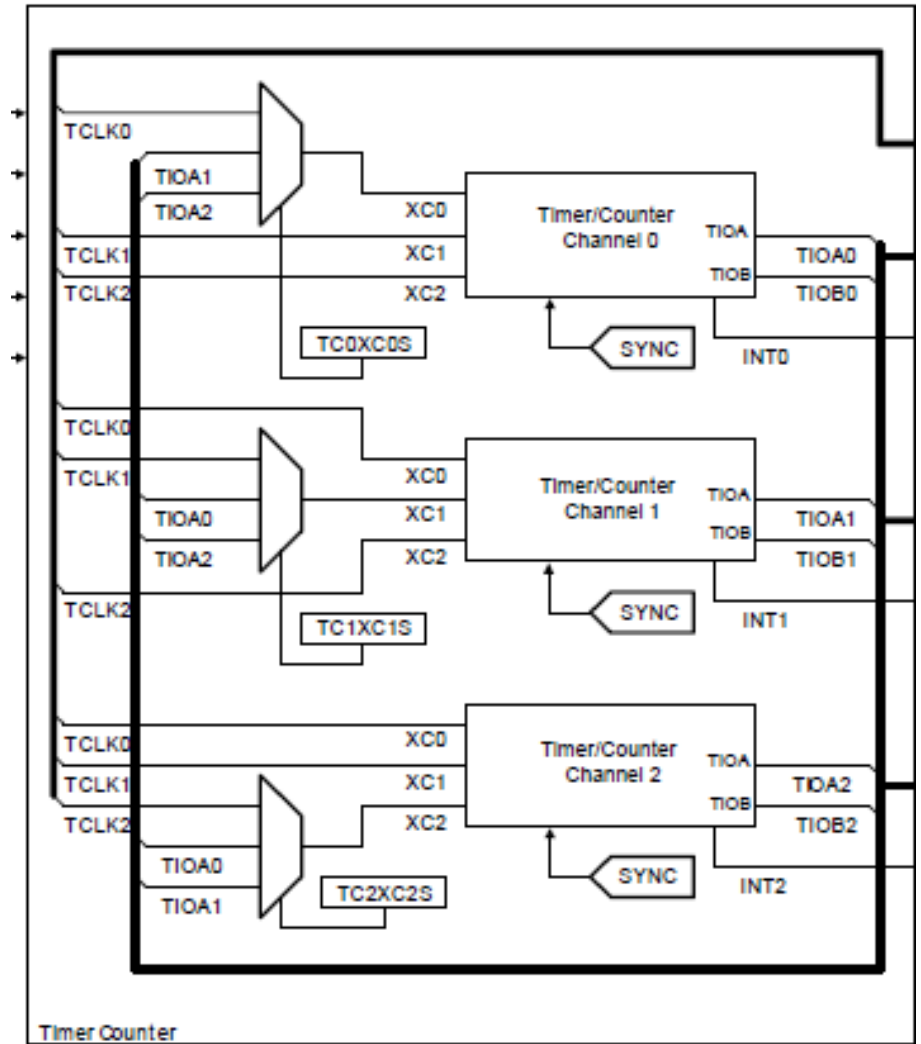
0x10	Channel 0 Counter Value	CV0	Read-only
0x14	Channel 0 Register A	RA0	Read/Write[1]
0x18	Channel 0 Register B	RB0	Read/Write[1]
0x1C	Channel 0 Register C	RC0	Read/Write

SAM4L PWM

- PWM peripheral is essentially built into Timer peripheral

- Timer peripheral also supports a "Waveform" mode where it generates an output GPIO signal

- One comparison point is low-to-high transition, another is high-to-low transition

Counter Value

0xFFFF

Counter cleared by compare match with RC

RC

RB

RA

Time

Waveform Examples

TIOB

TIOA
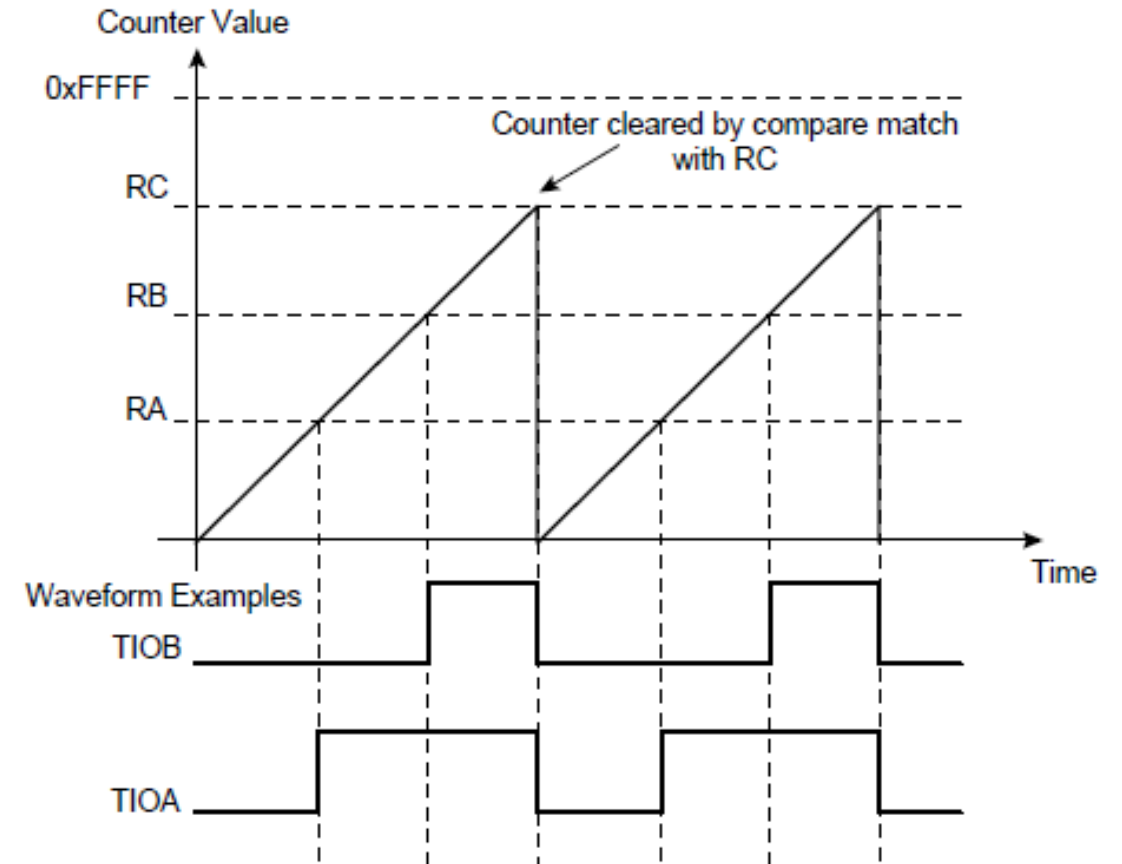
Texas Instruments MSP430 (MSP430FR59xx)

- Example of a *very* different microcontroller

- 16-bit custom architecture processor

- 2 kB RAM

- 64 kB FRAM
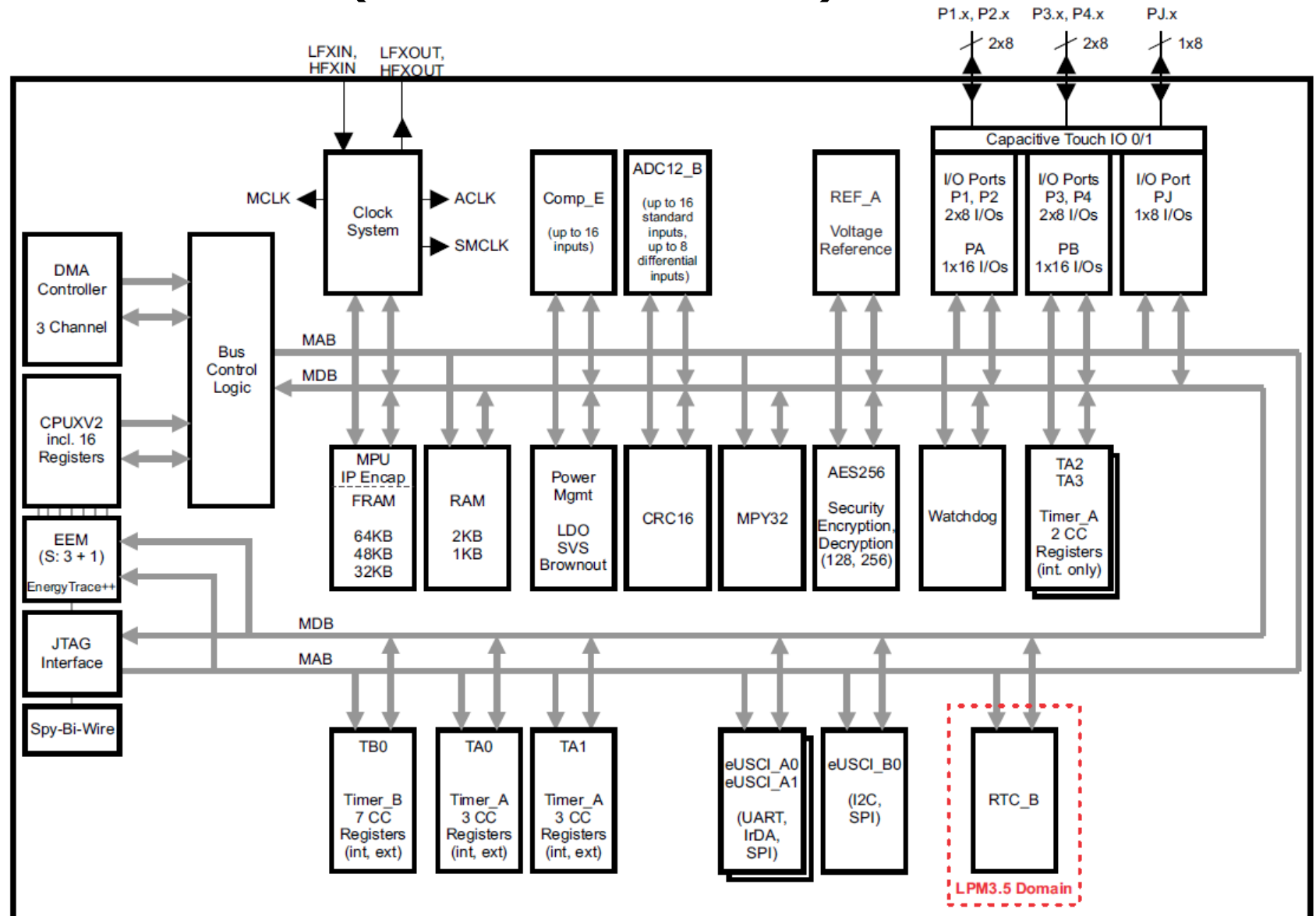
MSP430 GPIO register map

Table 12-3. Digital I/O Registers (continued)

Offset	Acronym	Register Name	Type	Access	Reset	Section
01h	P2IN or PAIN_H	Port 2 Input	Read only	Byte	undefined	Section 12.4.2
03h	P2OUT or PAOUT_H	Port 2 Output	Read/write	Byte	undefined	Section 12.4.3
05h	P2DIR or PADIR_H	Port 2 Direction	Read/write	Byte	00h	Section 12.4.4
07h	P2REN or PAREN_H	Port 2 Resistor Enable	Read/write	Byte	00h	Section 12.4.5
0Bh	P2SEL0 or PASEL0_H	Port 2 Select 0	Read/write	Byte	00h	Section 12.4.6
0Dh	P2SEL1 or PASEL1_H	Port 2 Select 1	Read/write	Byte	00h	Section 12.4.7
17h	P2SELC or PASELC_L	Port 2 Complement Selection	Read/write	Byte	00h	Section 12.4.8
19h	P2IES or PAIES_H	Port 2 Interrupt Edge Select	Read/write	Byte	undefined	Section 12.4.9
1Bh	P2IE or PAIE_H	Port 2 Interrupt Enable	Read/write	Byte	00h	Section 12.4.10
1Dh	P2IFG or PAIFG_H	Port 2 Interrupt Flag	Read/write	Byte	00h	Section 12.4.11

- Registers are a single byte in size
- Oddly, some have an "undefined" reset state

Example MSP430 GPIO register: OUT

- Example 8-bit register. Controls 8 pins in a GPIO "port"

12.4.3 PxOUT Register

Port x Output Register

Figure 12-3. PxOUT Register

7	6	5	4	3	2	1	0
			PxOUT				
rw	rw	rw	rw	rw	rw	rw	rw

Table 12-6. PxOUT Register Description

Bit	Field	Type	Reset	Description
7-0	PxOUT	RW	Undefined	Port x output When I/O configured to output mode: 0b = Output is low. 1b = Output is high. When I/O configured to input mode and pullups/pulldowns enabled: 0b = Pulldown selected 1b = Pullup selected

MSP430 Timer peripheral

- A choice of input clocks goes through dividers to run a 16-bit timer
  - Five total timers available on the system
  - Some can trigger external output pins, some are only internal
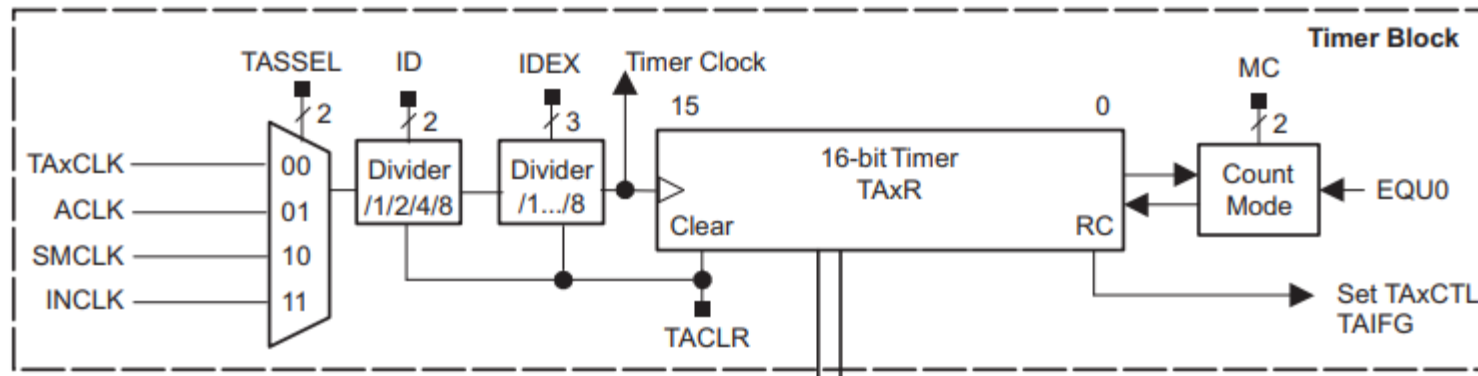
MSP430 PWM

- The MSP430 Timer peripheral handles PWM as well!

25.2.3.1 Up Mode

The up mode is used if the timer period must be different from 0FFFFh counts. The timer repeatedly counts up to the value of compare register TAxCCR0, which defines the period (see Figure 25-2). The number of timer counts in the period is TAxCCR0 + 1. When the timer value equals TAxCCR0, the timer restarts counting from zero. If up mode is selected when the timer value is greater than TAxCCR0, the timer immediately restarts counting from zero.
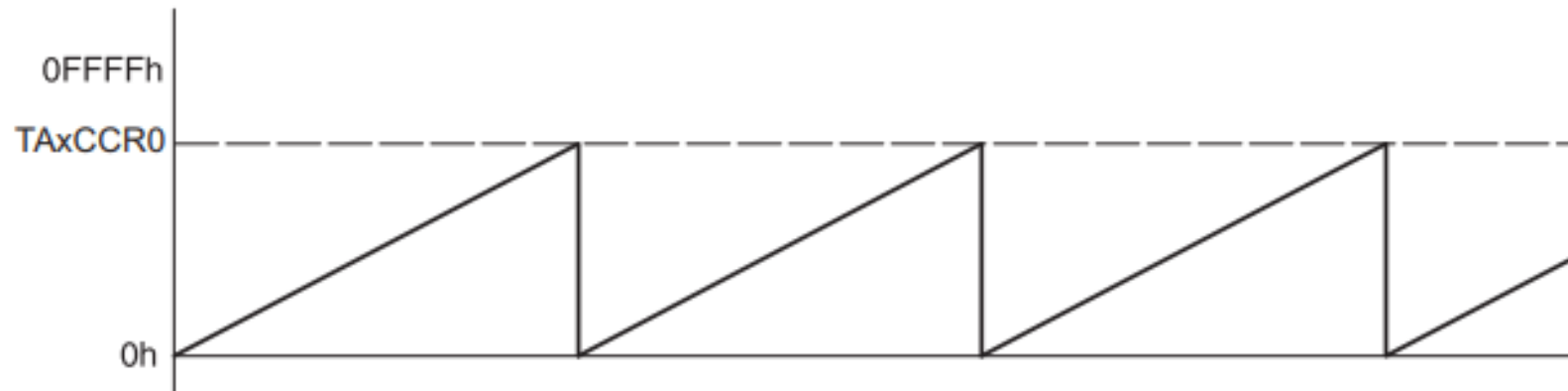

Figure 25-2. Up Mode

Outline

- What haven't we talked about?
 - Microbit
  - nRF52833

- **Other hardware systems**
  - Other microcontrollers
  - **Microprocessors**
  - FPGAs

Microprocessor system design

- What about microprocessors?
  - Similar idea, but a LOT more stuff

- Various peripherals, which might not be focused on sensor I/O
  - Less ADCs and I2C more Ethernet and Graphics

- External memory busses
  - Microprocessors expect external memory to exist on the system
  - Which means that they need pins for an external memory bus

Beaglebone

- Cheap, single board computer
  - Like Raspbery Pi
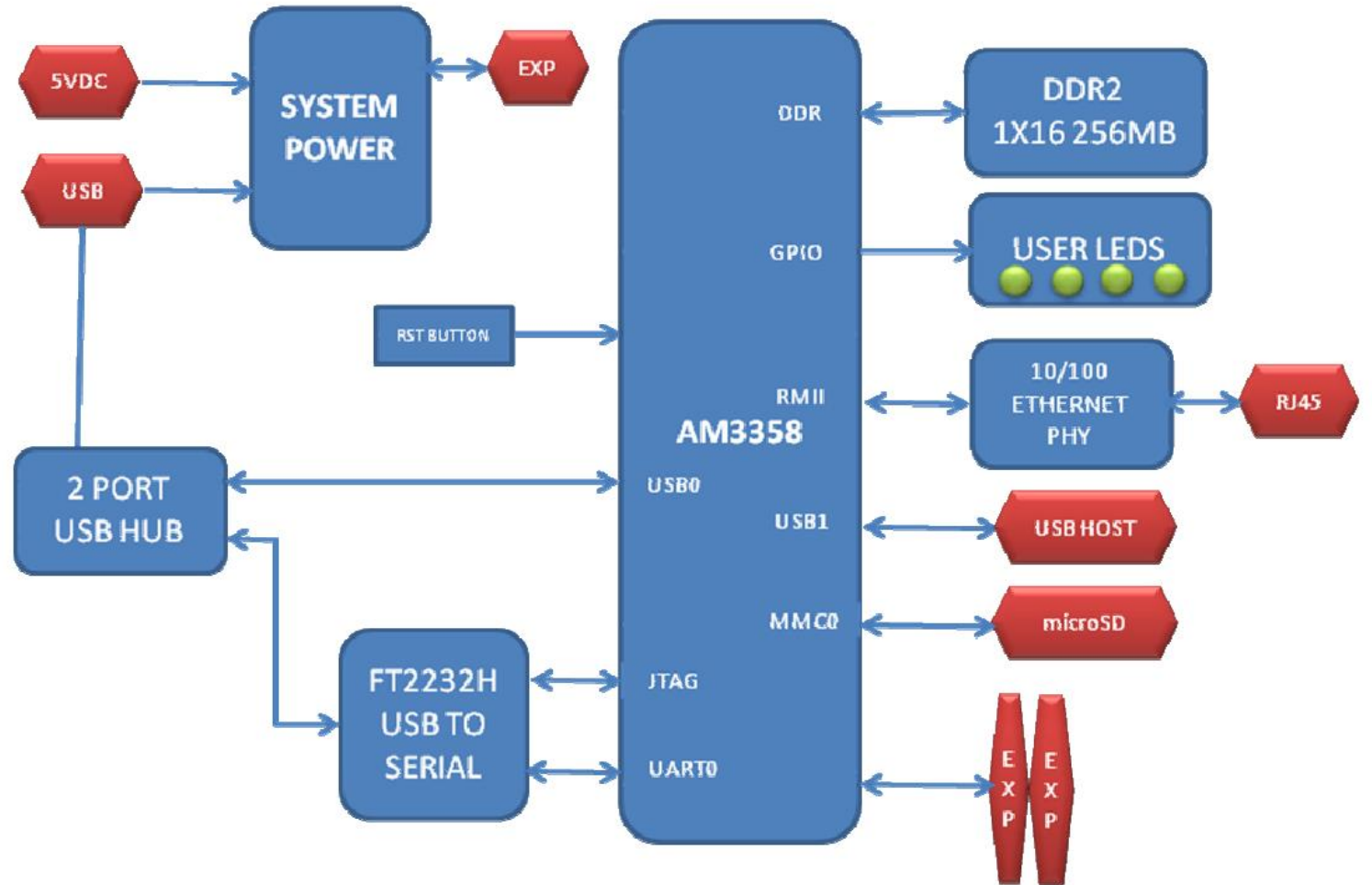
AM3358 processor

- ARM Cortex A8
  - 32-bit processor
  - 1 GHz clock
  - 300 pins!!

- Internal memory
  - 176 kB ROM
  - 64 kB RAM

- External memory
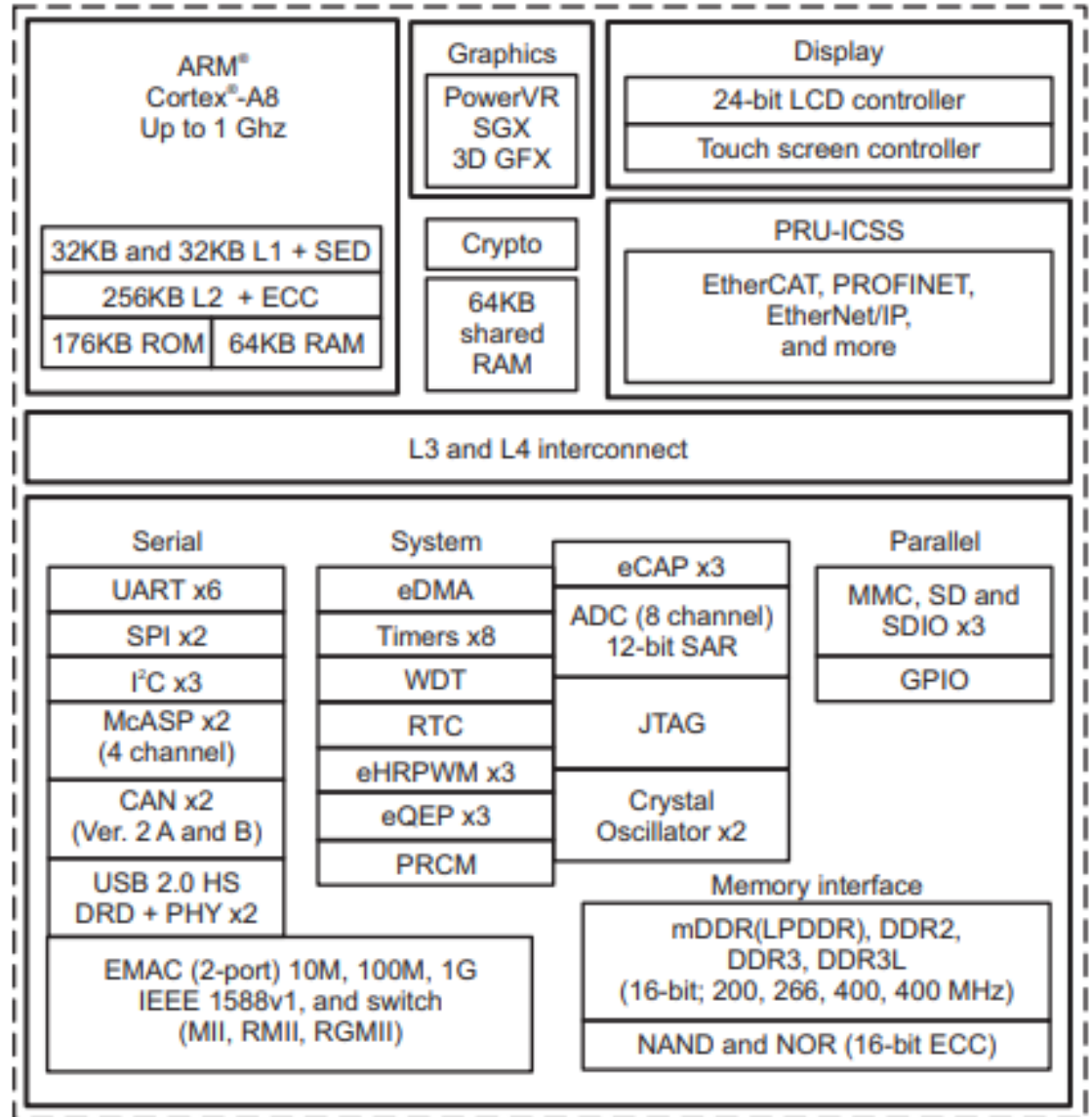 - DDR2/3, 400 MHz RAM
  - NAND or NOR Flash

Figure 1-1. AM335x Functional Block Diagram

AM3358 peripherals

- Advanced peripherals
  - Graphics
  - Ethernet
  - USB, CAN

- Regular peripherals
  - GPIO, Timers
  - ADC
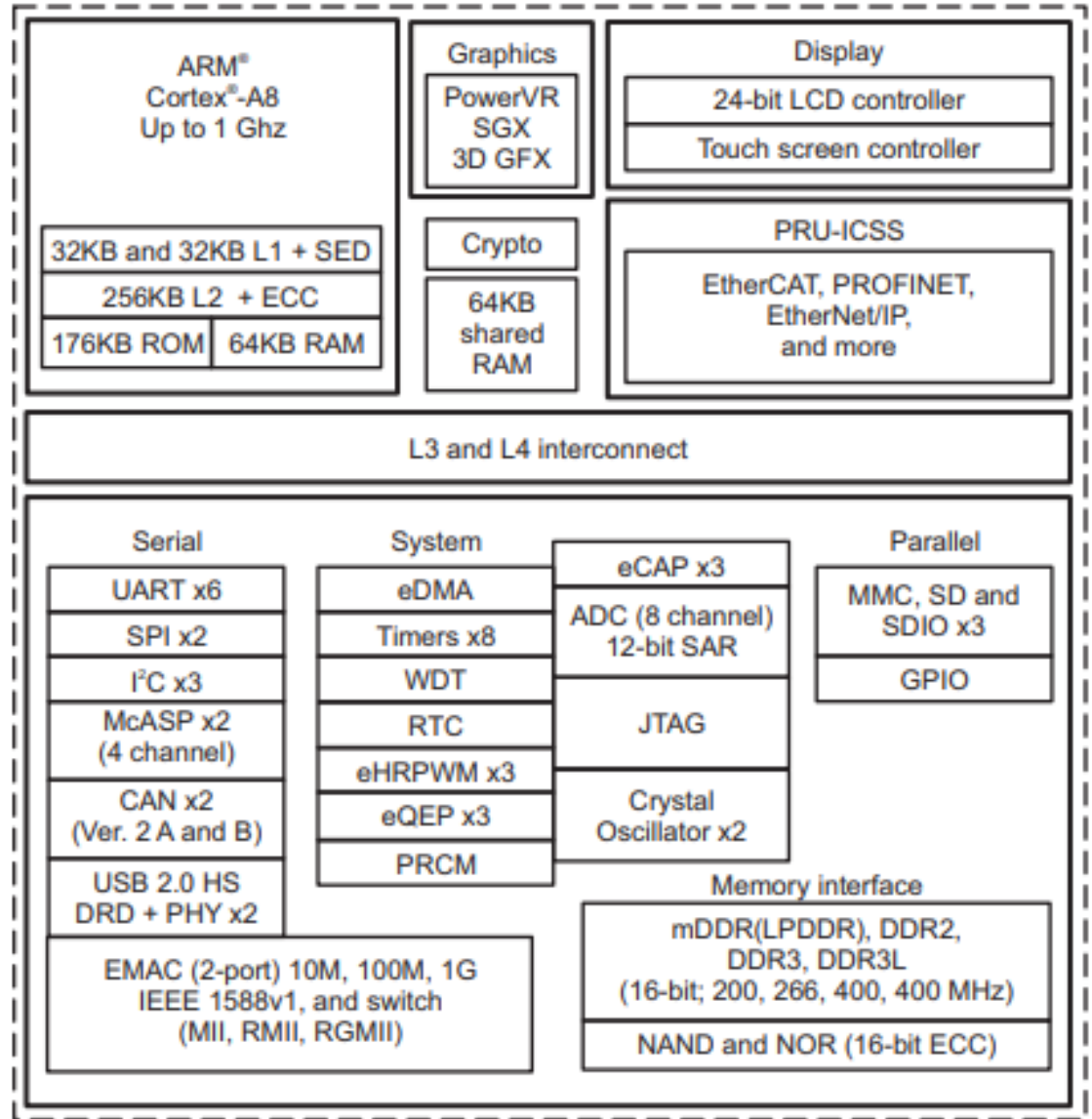 - UART, SPI, I2C

Figure 1-1. AM335x Functional Block Diagram

How much bigger is a microprocessor?

- nRF52833 datasheet:
 - 600 pages total

- AM3358 datasheet:
  - 250 pages of electrical information
  - 5000 pages of peripheral information

  - And it's not that the basic peripherals are much larger
    - UART: 60 pages
    - I2C: 20 pages
    - Four different timers at ~30 pages each

Memory interface - DDR

- Double Data Rate (DDR)

- Address bits
  - 17 pins (3 bank, 14 address)
  - Bank + Address selects a 2 kB chunk out of 2 GB

- Data bits
  - 16 pins

- Clocks, Chip select

- Various sequencing signals

Pinout on the Beaglebone
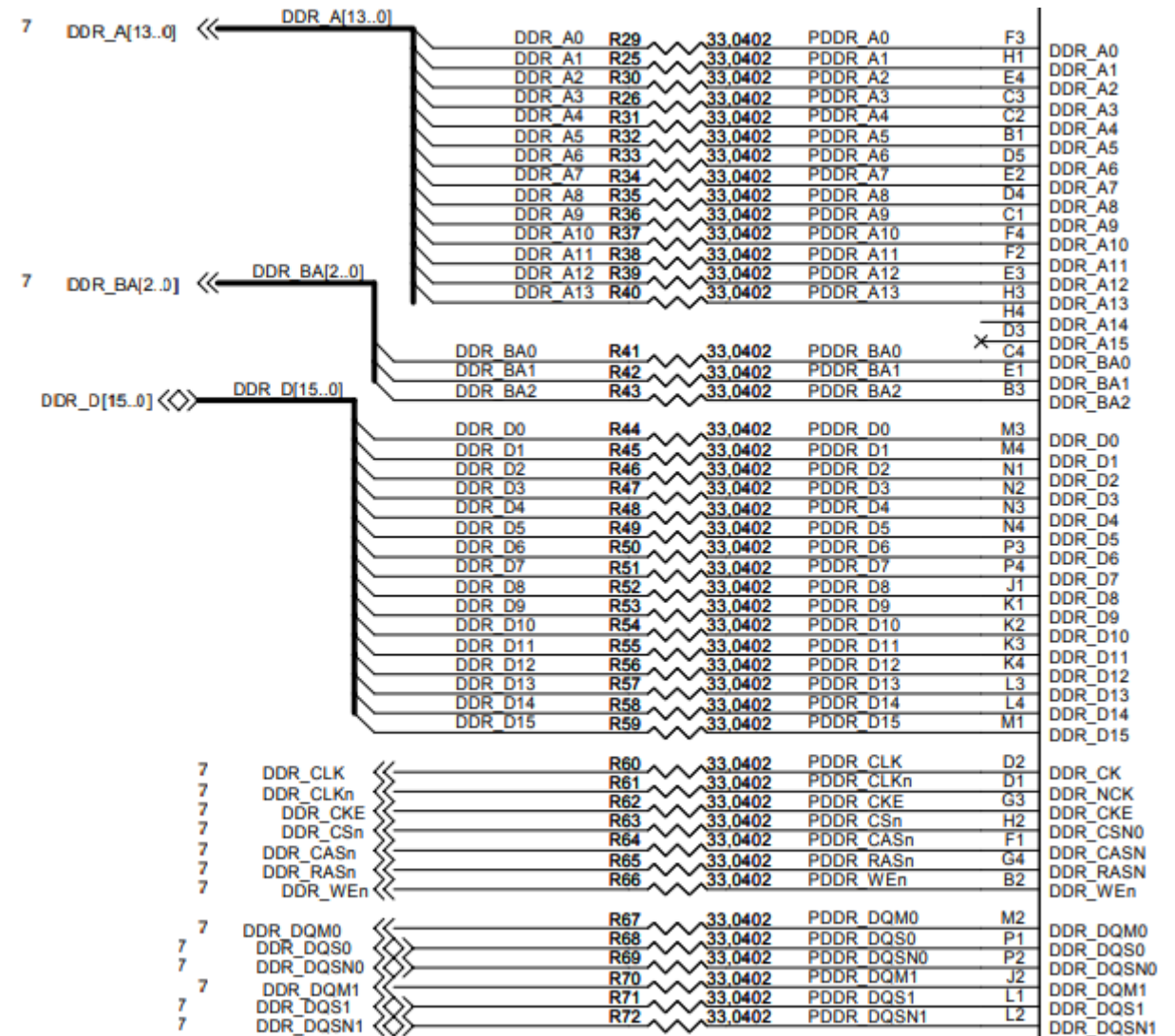
Break + Question

- Wires (traces) for high-speed busses like memory must be match, to ensure that all are exactly the same length.

  Why?

Break + Question

- Wires (traces) for high-speed busses like memory must be match, to ensure that all are exactly the same length.
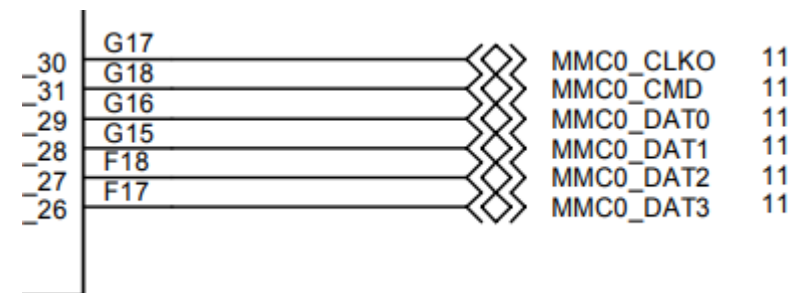
  Why?

- Speed of electricity in the wires matters

- If one wire is a foot longer than the others, its signal arrives 1 ns late
  - Even fractions of nanoseconds could matter for high-speed communication

Flash interface - MMC

- Multi-media Card interface
    - Used to communicate with SD Cards
    - Also for eMMC Flash chips (which act like SD Cards, but soldered in place)

- For SD cards SPI can usually be used

- For eMMC 4-bit version of interface
    - Clock
    - Command
   - Data 0, 1, 2, 3

Pinout on the Beaglebone

|		
_30	G17 / G18	MMC0_CLKO 11
_31	G18	MMC0_CMD 11
_29	G16	MMC0_DAT0 11
_28	G15	MMC0_DAT1 11
_27	F18	MMC0_DAT2 11
_26	F17	MMC0_DAT3 11

Ethernet interface - RMII

- Reduced Media-Independent Interface (RMII)

Signal name	Description
REF_CLK	Continuous 50 MHz reference clock
TXD0	Transmit data bit 0 (transmitted first)
TXD1	Transmit data bit 1
TX_EN	When high, clock data on TXD0 and TXD1 to the transmitter
RXD0	Receive data bit 0 (received first)
RXD1	Receive data bit 1
CRS_DV	Carrier Sense (CRS) and RX_Data Valid (RX_DV) multiplexed on alternate clock cycles. In 10 Mbit/s mode, it alternates every 10 clock cycles.
RX_ER	Receive error (optional on switches)
MDIO	Management data
MDC	Management data clock.

Pinout on the Beaglebone
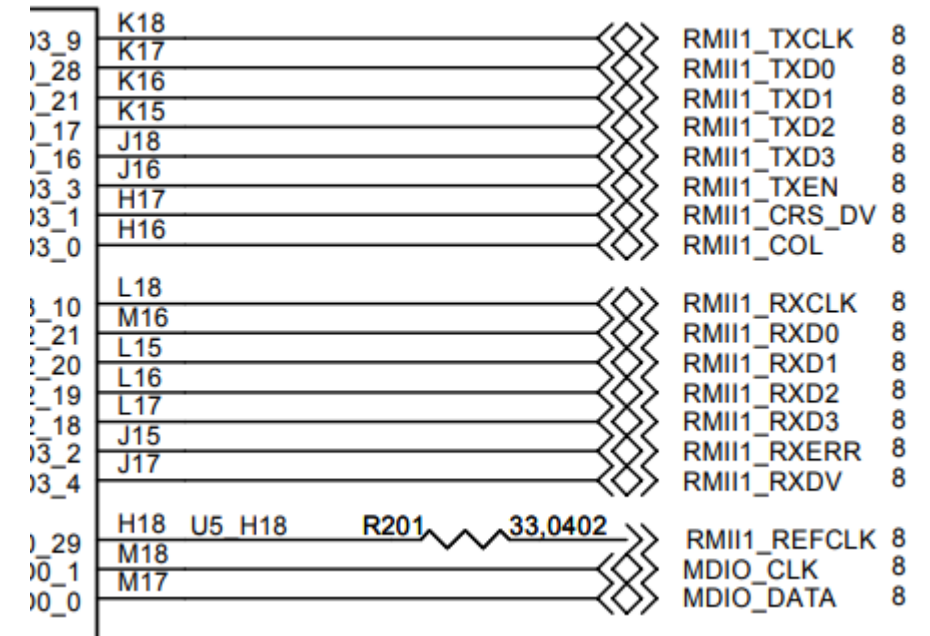
Other processor pinouts on the Beaglebone
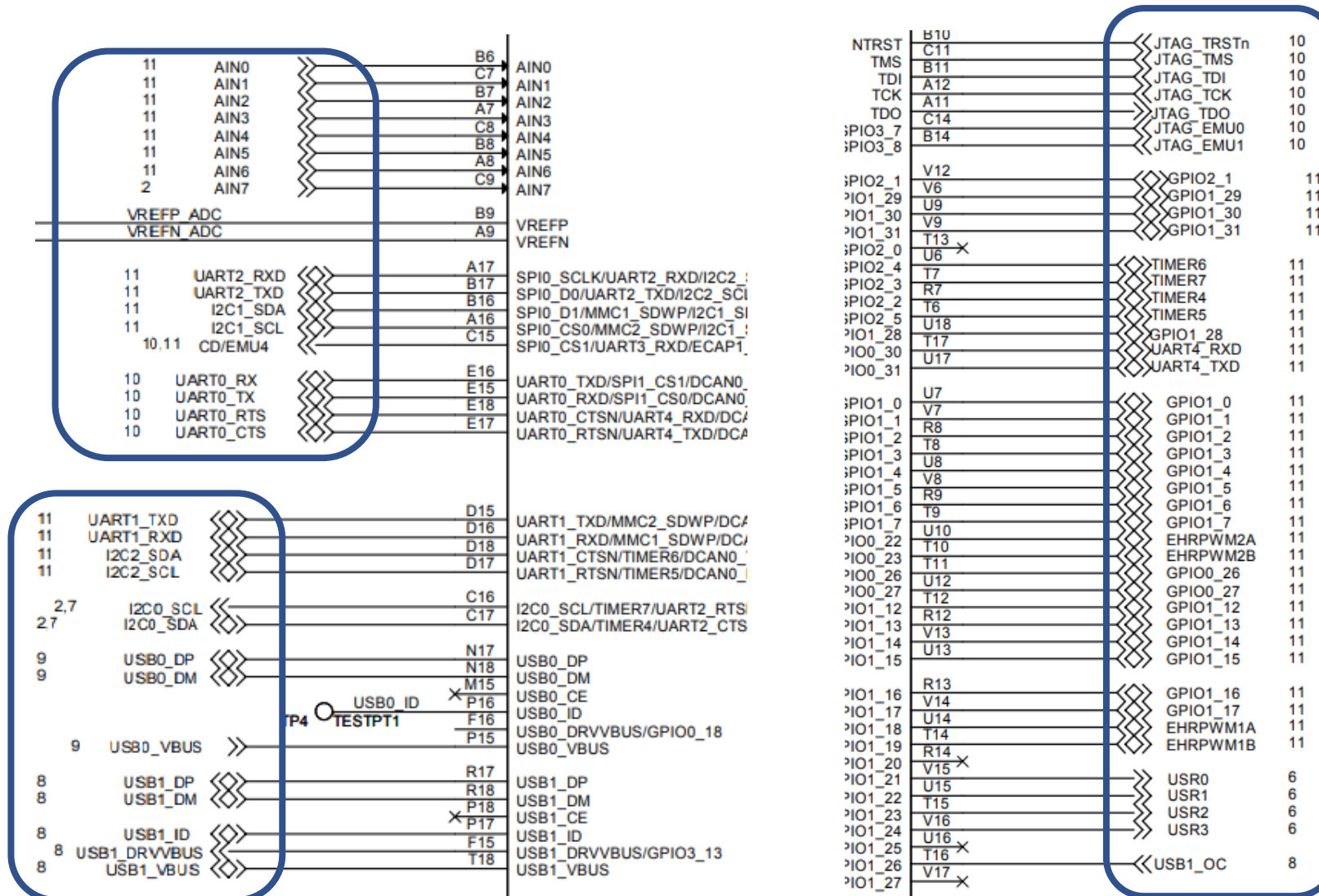
- Lots of acronyms that we recognize!

- Analog inputs
- UARTs
- I2C
- GPIOs
- Timers

Resources on the Beaglebone and its microprocessor

- https://beagleboard.org/bone-original
- https://beagleboard.org/static/beaglebone/BEAGLEBONE_SCHEMA3.pdf
- https://transistor-man.com/files/emu/beaglebone_hardware/BONE_SRM.pdf

- https://www.ti.com/product/AM3358

Outline

- What haven't we talked about?
  - Microbit
  - nRF52833

- **Other hardware systems**
  - Other microcontrollers
 - Microprocessors
  - **FPGAs**

FPGA systems

- When else might you run into large busses like memory?

- On an FPGA!
  - Field-Programmable Gate Array (FPGA)

  - Configurable hardware
    - Engineers write a description of the hardware (Verilog or VHDL)
    - FPGA implements that hardware

  - Much faster than software, but still slower than a custom chip

Combined FPGA + Microcontroller

Busses: AHB vs APB

AHB to APB Bridge
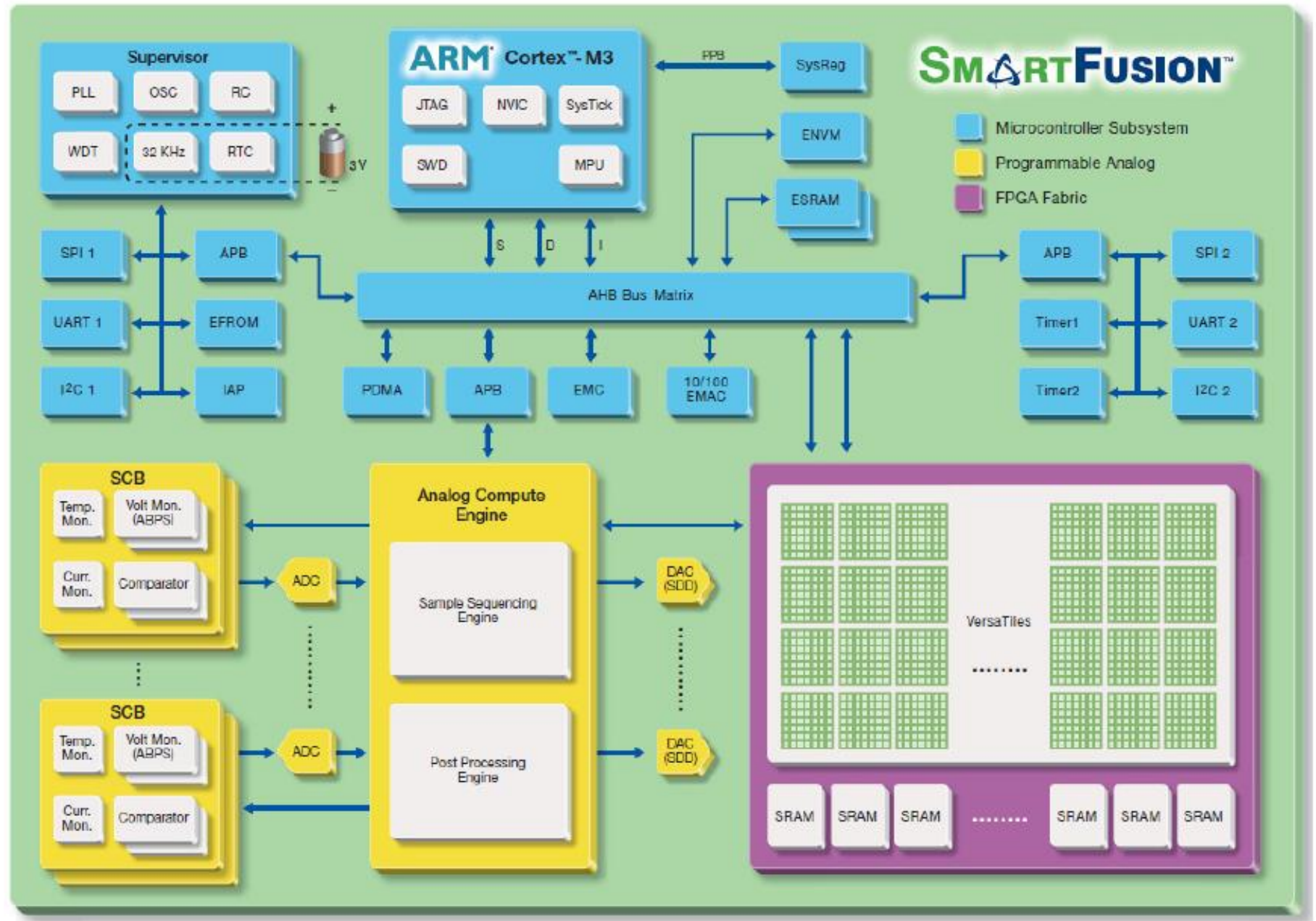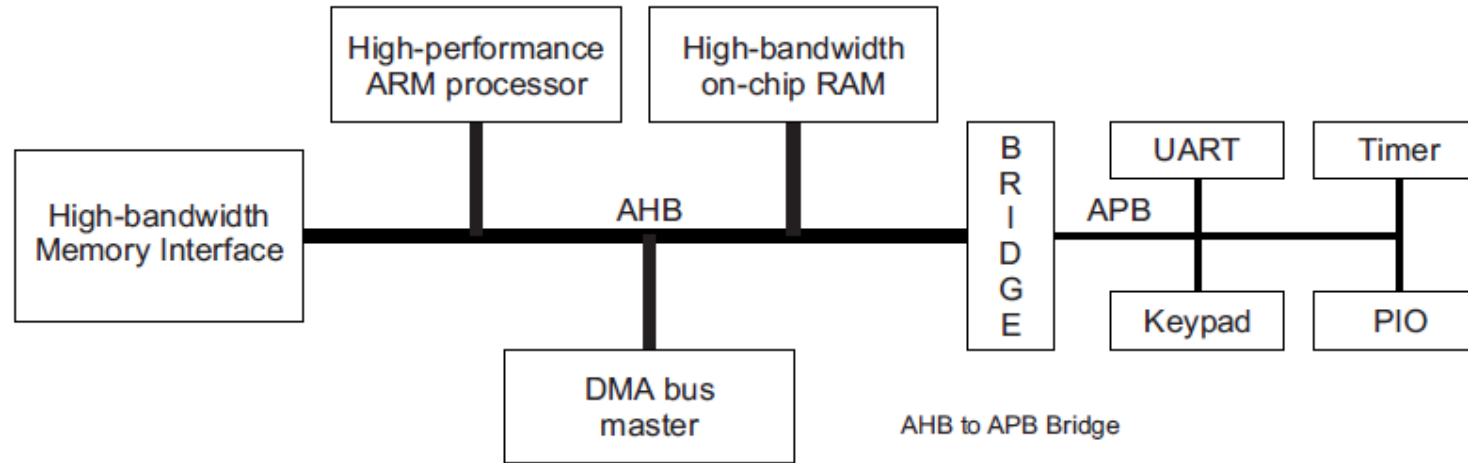
- Advanced High-performance Bus (AHB)
  - Multiple bus controllers
  - Pipelined operation
  - Burst transfers of data
 - Used for high-speed memory operations

- Advanced Peripheral Bus (APB)
  - Simpler interface
  - Lower power
  - Used for most peripheral communication
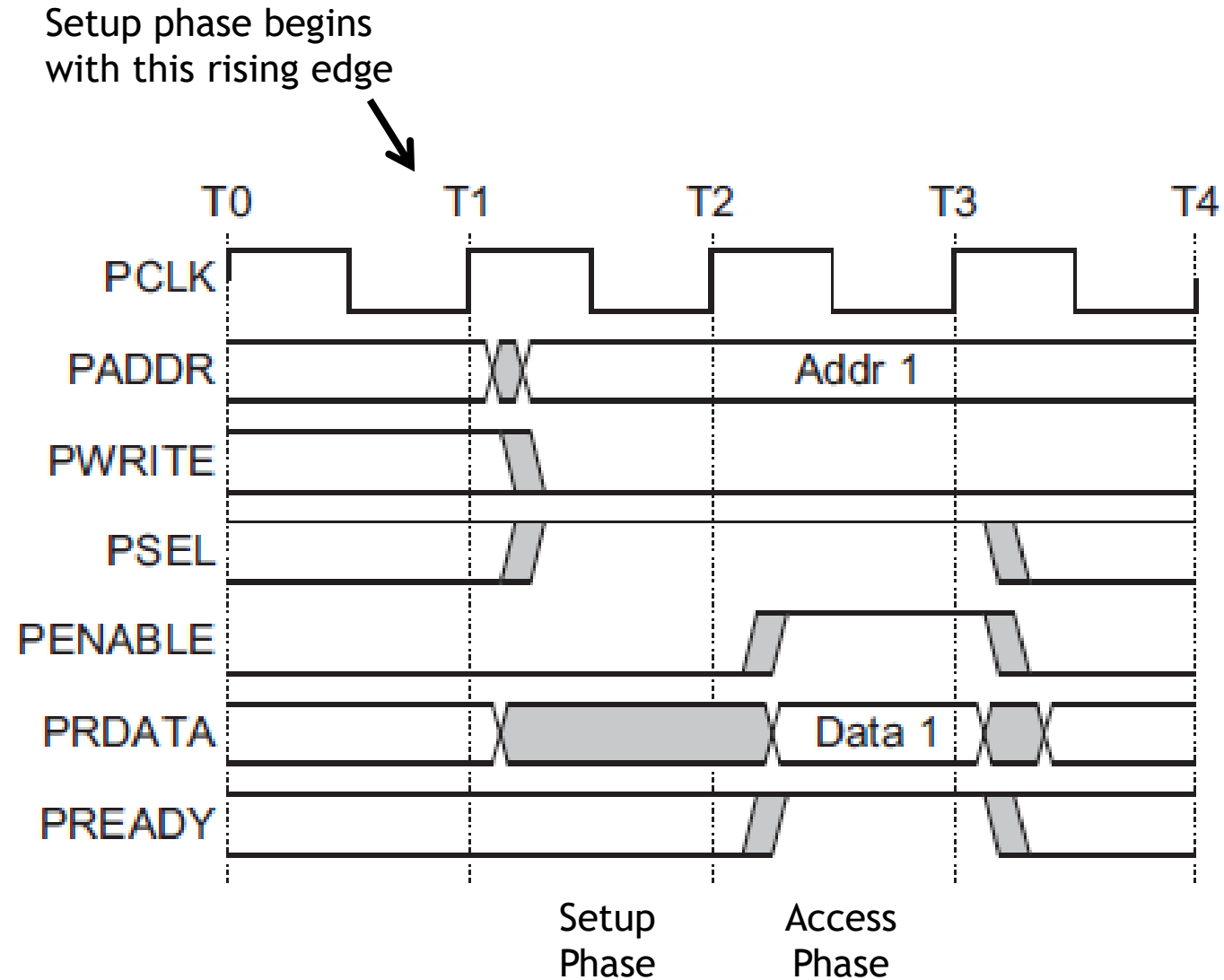
APB usage

- Bus has three states
    - IDLE
        - Nothing is happening
        - Default

    - SETUP
        - A transfer of data is about to happen
        - Address and configuration signals are set

    - ACCESS
        - Data is actually transferred over the bus

Example APB write operation
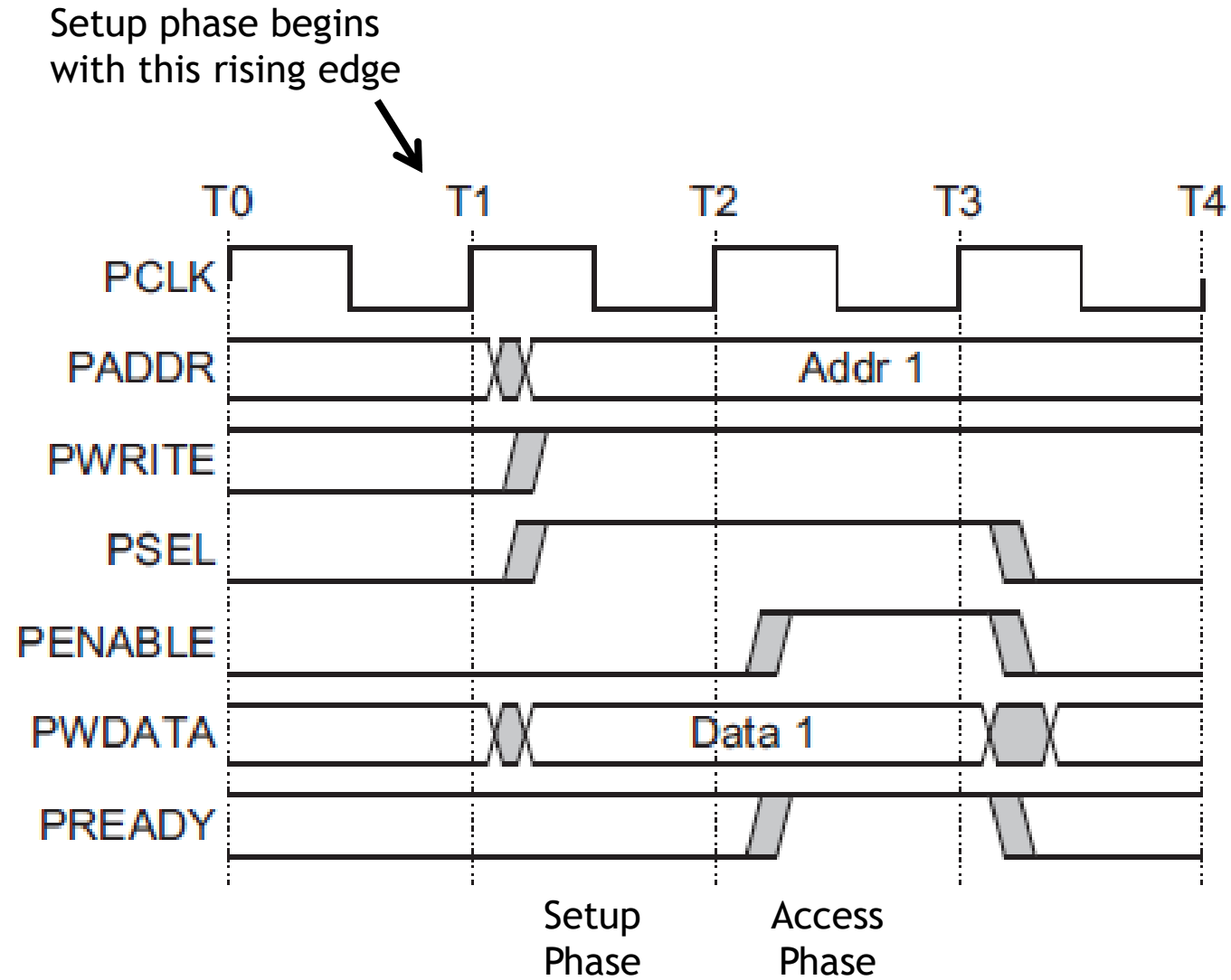
Example APB read operation

With an FPGA you can create your own peripherals

- Busses can be exposed to the FPGA for it to use

- Example use case
  - Custom camera with selectable pixel lines and an ADC
  - Don't want to manually control it with software (too slow)

  - Create a hardware peripheral on the FPGA that takes software commands like "grab a frame" and implements them

Example register map

- Verilog HDL that describes a register map

- Reads in the address from the bus to determine which register is written to

- Modifies internal registers based on the bits of the data

```verilog
// Interact with bus
if (bus_write) begin
    case (PADDR[7:0])
        `GLOB_START: begin
            cam0_frame_capture_start <= PWDATA[0];
            cam1_frame_capture_start <= PWDATA[1];
        end
        `CAM0_FRAMEMASK: begin
            cam0_mask_write_enable <= 1;
            cam0_mask_addr <= {PWDATA[30:24], PWDATA[18:16]};
            cam0_mask_data <= PWDATA[15:0];
        end
        `CAM0_SETTINGS1: begin
            cam0_vref_value    <= PWDATA[29:24];
            cam0_config_value  <= PWDATA[21:16];
            cam0_nbias_value   <= PWDATA[13:8];
            cam0_aobias_value  <= PWDATA[5:0];
        end
        `CAM0_SETTINGS2: begin
            cam0_val_offset <= PWDATA[27:16];
            cam0_vsw_value  <= PWDATA[15:8];
            cam0_hsw_value  <= PWDATA[7:0];
        end
```

Performing peripheral operations

- On every clock edge, check status registers

- If they are active, iterate through a state machine
  - Possibly many states to make the actual operation occur

```verilog
`IDLE: begin
    if (frame_capture_start) begin
        mask_pixel_col_nxt = 0;
        mask_pixel_row_nxt = 0;
        main_state_nxt = `CAPTURE;
    end else begin
        frame_capture_done_nxt = 0;
        main_state_nxt = `IDLE;
    end
end
```

Outline

- What haven't we talked about?
  - Microbit
  - nRF52833

- Other hardware systems
  - Other microcontrollers
  - Microprocessors
  - FPGAs