COPYRIGHT

DISCLAIMER

This content is provided without warranty or representation of any kind. The use of the content is entirely at your own risk and Stellenbosch University (SU) will have no liability directly or indirectly as a result of this content.

The content must not be assumed to provide complete coverage of the particular study material. Content may be removed or changed without notice.

The video is of a recording with very limited post-recording editing. The video is intended for use only by SU students enrolled in the particular module.

UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

100
1918 · 2018

forward together · saam vorentoe · masiye phambili

Computer Systems / Rekenaarstelsels 245 - 2020

Lecture 11

System Bus
Stelselbus

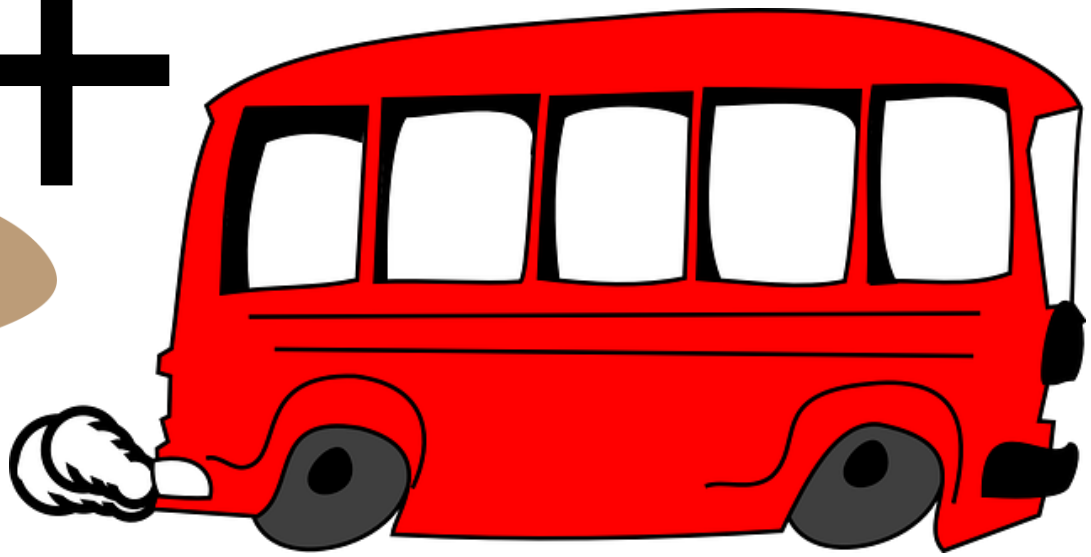
Dr Rensu Theart & Dr Lourens Visagie

Lecture Overview

- Computer bus
- System bus
- Address decoding
- AHB
- Cortex-M4 system bus

What is a computer bus? / Wat is ‘n rekenaarbus?

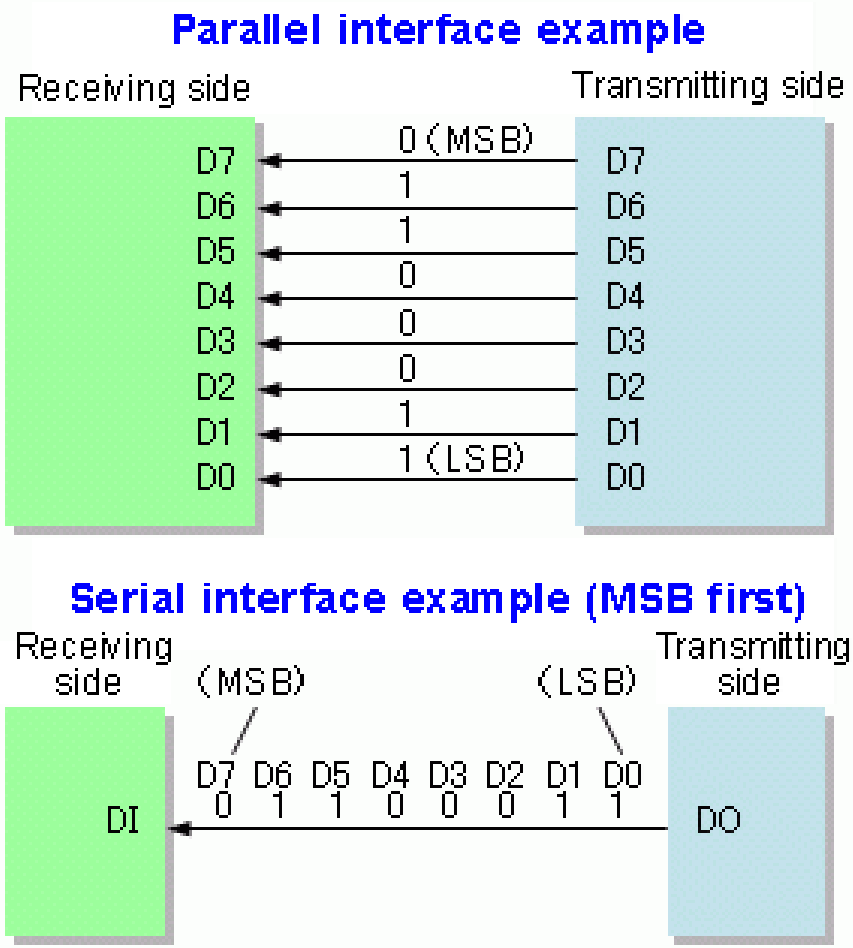
+



???

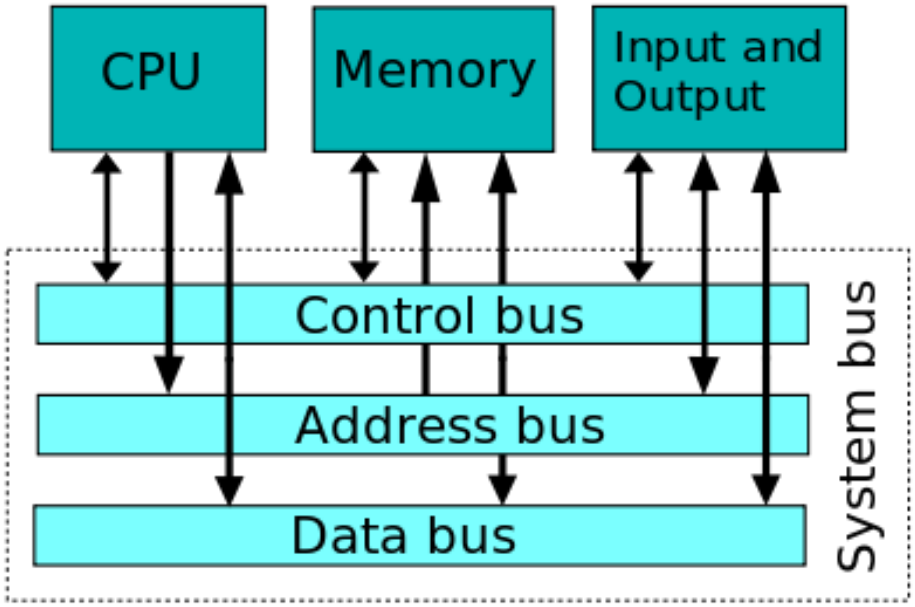
The Computer Bus / Die Rekenaarbus

- A **bus** is a communication system that transfers data between components inside a computer, or between computers.
- Generic term used for serial and parallel communications, with provision for multiple sources/destinations using the same physical connection (wire/PCB track/conductor inside IC).
- **Serial** communications: Single line/signal/wire/channel that is varied over time to transfer multiple bits of information.
- **Parallel** communications: Multiple lines/signals/wires/channels (also varying over time) that can transfer multiple bits of information at a single time instant.



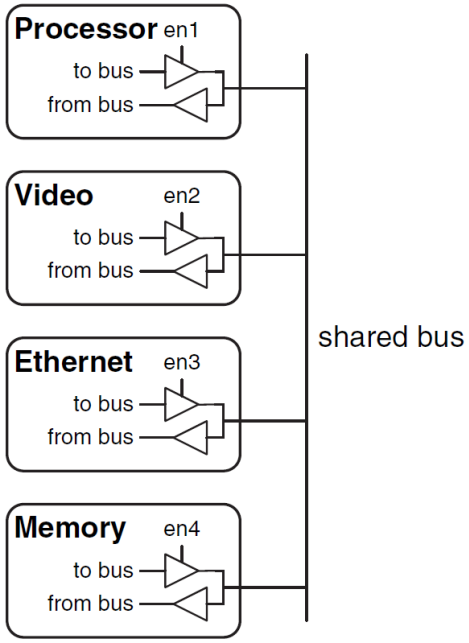
The System Bus / Die Stelselbus

- The **System bus/Internal bus** – connects CPU and memory and peripherals
- A parallel bus with
  - Address lines
  - Data Lines
  - Control lines
- Key hardware elements of the system bus
  - Tri-state buffer
  - Address decoder (binary decoder)

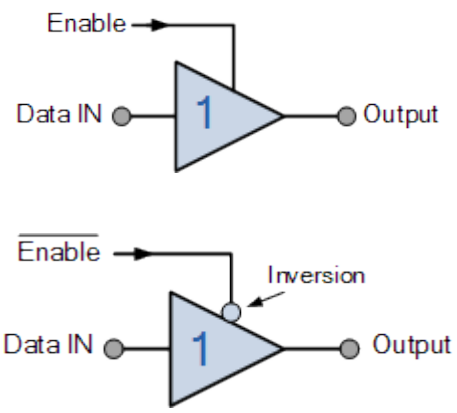


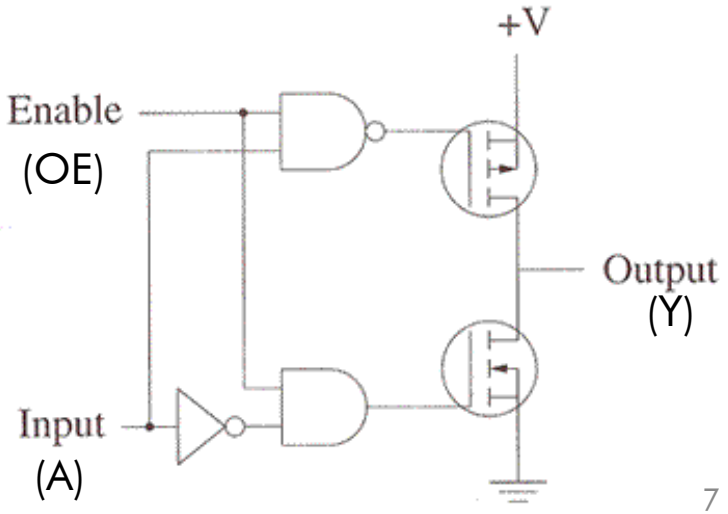
The System Bus / Die Stelselbus

- Digital **Buffers** can be used to isolate other gates or circuit stages from each other preventing the impedance of one circuit from affecting the impedance of another.
- A **Tri-state buffer** is type of digital buffer circuit whose output can be electronically disconnected from its output circuitry when required.
- Tri-state buffer allows multiple devices to communicate using the same bus – a key element in the way data is transferred between CPU and memory.



A	OE	Y
0	1	0
1	1	1
0	0	Hi-impedance
1	0	



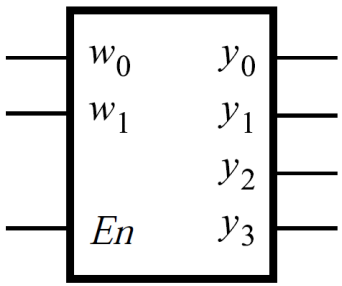


Address Decoding / Adres dekodering

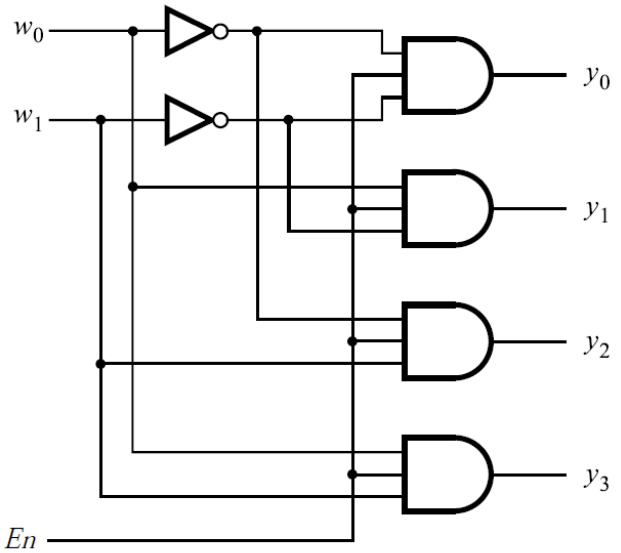
- Binary Decoder (2:4)
- Produces a **one-hot** encoded output, where only a single output line is HIGH at a time, based on the binary value at the input.

En	w_1	w_0	y_0	y_1	y_2	y_3
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1
0	x	x	0	0	0	0

(a) Truth table

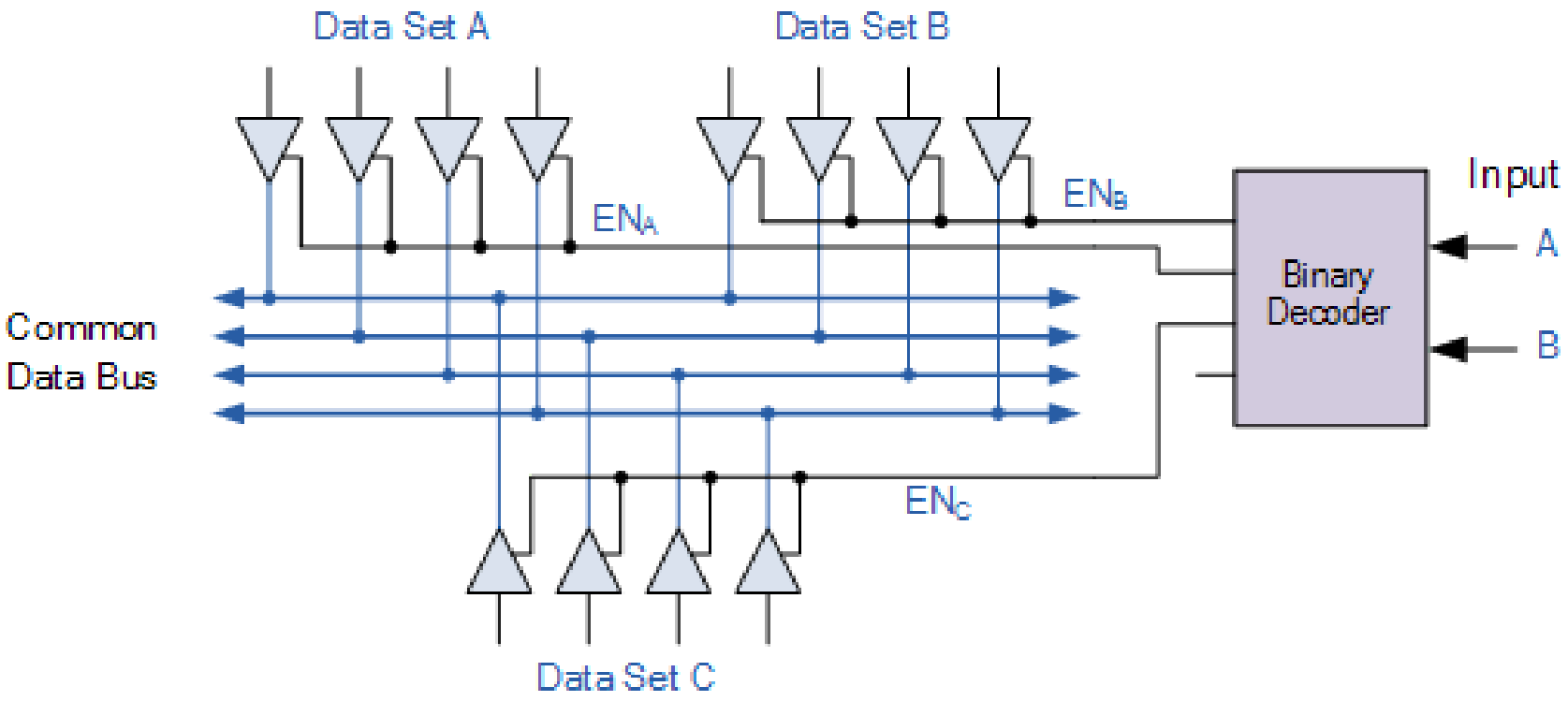


(b) Graphical symbol



Address Decoding / Adres dekodering

- A bi-directional bus allows communication in both directions.
  - Some form of synchronization is required.
- Binary decoder is used to enable certain tri-state buffer only.
  - This prevents **bus contention**



Address Decoding / Adres dekodering

- The figure shows the hardware needed to support two memory-mapped I/O devices.
- An **address decoder** determines which device communicates with the processor.
- It uses the *Address* and *MemWrite* signals to generate control signals for the rest of the hardware.
- The *ReadData* multiplexer selects between memory and the various I/O devices.
- **Write-enabled registers** hold the values written to the I/O devices.

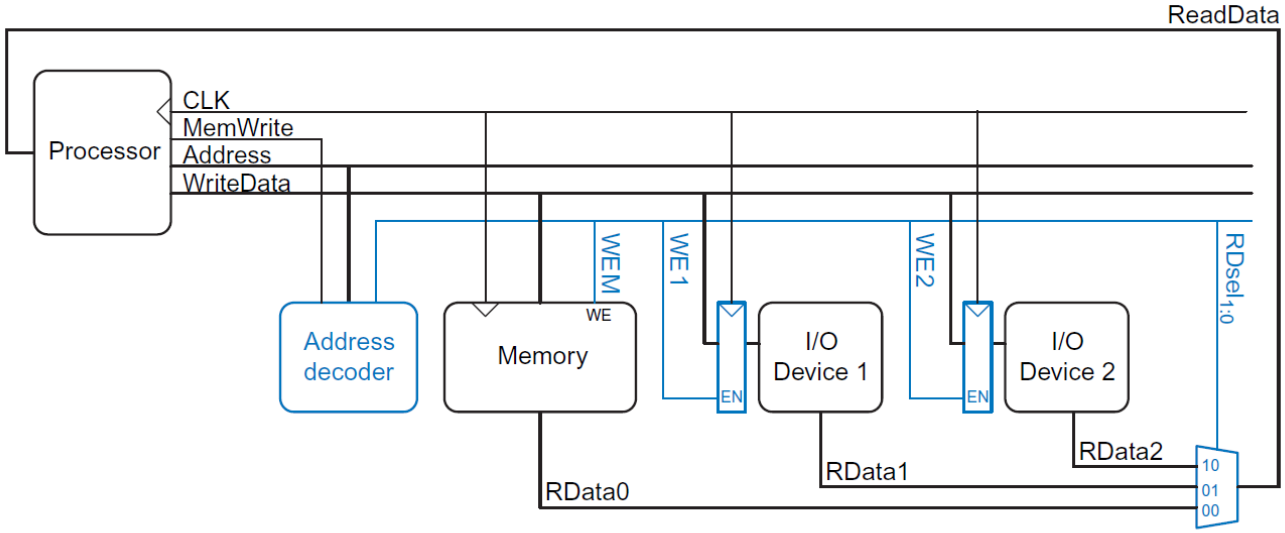
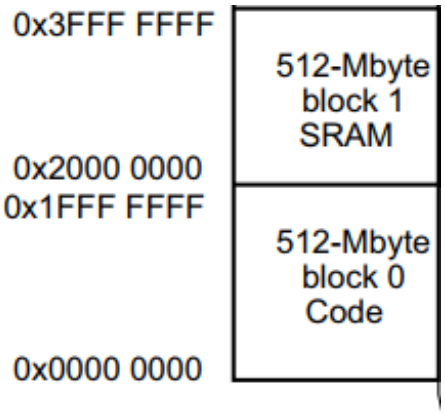
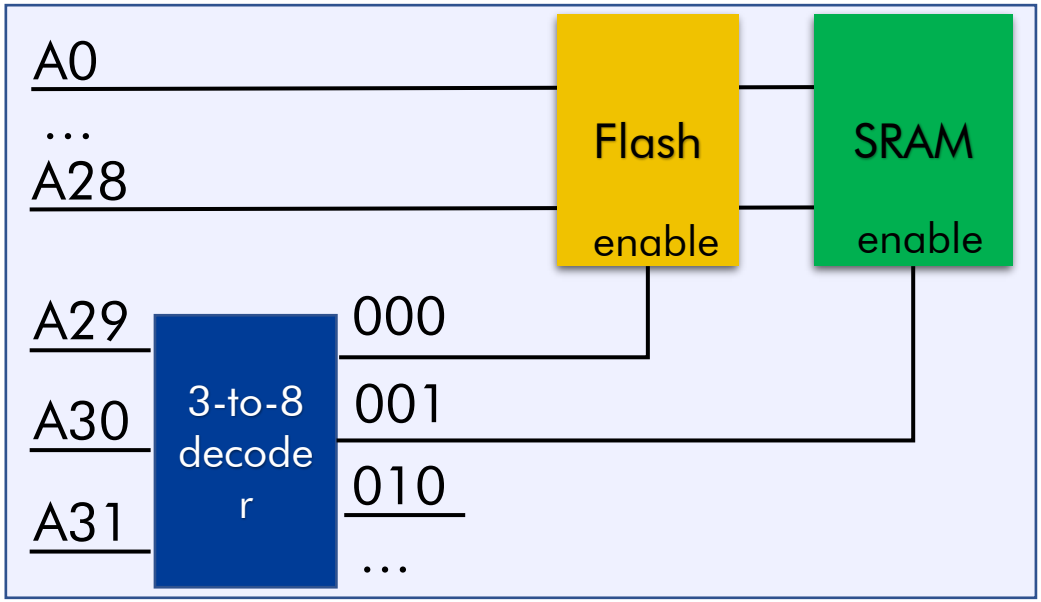


Figure e9.1 Support hardware for memory-mapped I/O

Address Decoding / Adres dekodering

Address decoding to select between different devices





Flash (code) memory addresses

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

SRAM addresses

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

AHB

- A bus interface connects processors to memory and/or peripherals.
- In general, a bus interface supports one or more bus **masters** that can initiate read or write requests to the bus and one or more **slaves** that respond to the requests.
  - Processors are normally masters and memory and peripherals are slaves.
- The **Advanced Microcontroller Bus Architecture (AMBA)** is an open standard bus interface for connecting components on a chip.
  - Introduced by ARM in 1996, it has developed through multiple revisions to boost performance and features and has become a de facto standard for embedded microcontrollers.
- The **Advanced High-performance Bus (AHB)** is one of the AMBA standards.
  - AHB-Lite is a simplified version of AHB that supports a single bus master.
- AHB is an example of a **point-to-point read bus.**
  - In contrast with older bus architectures that use a single shared databus where each slave accesses the bus via a tristate driver.
- Using point-to-point links between each slave and the read multiplexer allows the bus to run faster and avoids wasting power when one slave turns on its driver before another has turned off.

AHB

- Here is a simplified version of the AHB bus (AHB-Lite).
- Observe that the bus is very similar to the one from Figure e9.1 except that the names have changed.
- The master provides a **synchronous clock (HCLK)** to all of the slaves and can reset the slaves by asserting **HRESETn** low.
- The master sends an address. The address decoder uses the most significant bits to generate the **HSEL** signal selecting which slave to access, and the slaves use the least significant bits to define the memory location or register.

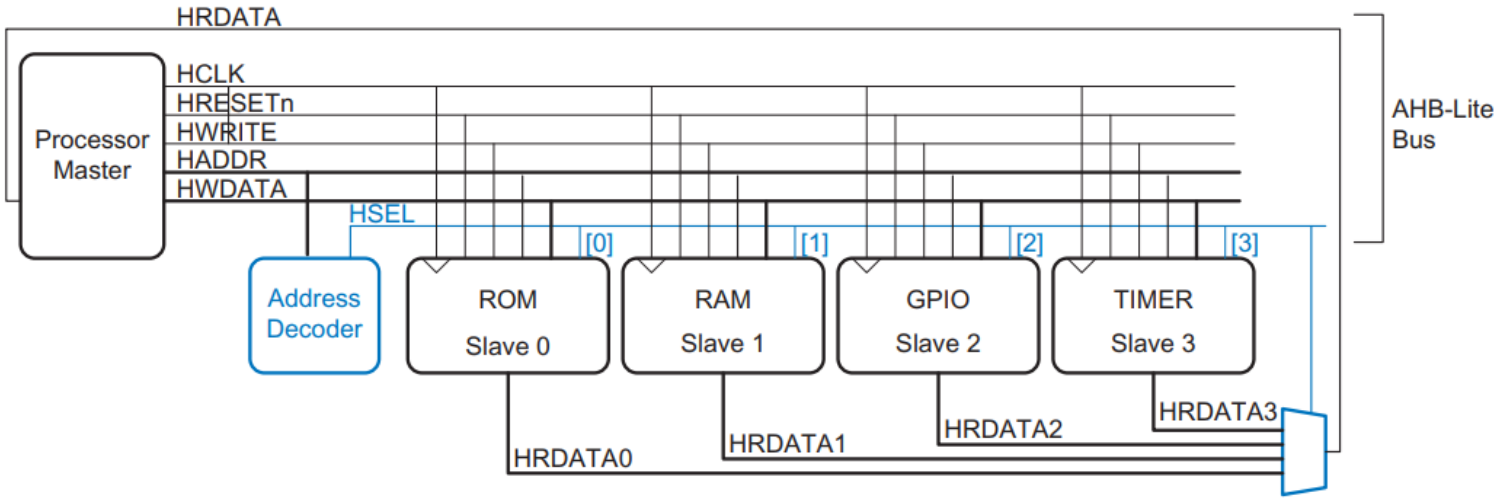


Figure e9.45 AHB-Lite bus

AHB

- The master sends **HWDATA** for writes.
- Each slave reads onto its own **HRDATA**, and a multiplexer chooses the data.
- For writes, the master raises **HWRITE** and sends the 32-bit HWDATA to write.
- For reads, the master lowers HWRITE and the slave responds with 32-bit HRDATA.
- Therefore the master can send an address on one cycle and writes or reads data on the subsequent cycle.
- Slaves can deassert **HREADY** to indicate that they need multiple clock cycles to respond, or can assert **HRESP** to indicate an error.

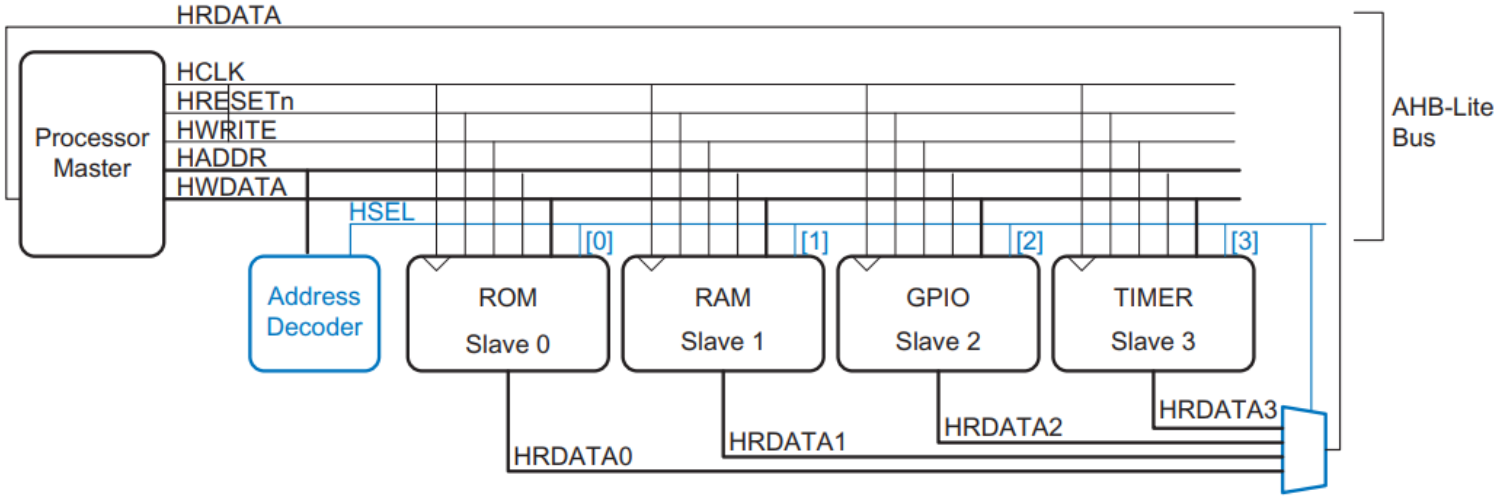


Figure e9.45 AHB-Lite bus

AHB

- Transfers can overlap so that the master can send the address of the next transfer while reading or writing data for the current transfer.
- This figure illustrates the timing of the bus for a write followed immediately by a read.
- Observe how the data lags one cycle behind the address and how the two transfers partially overlap.

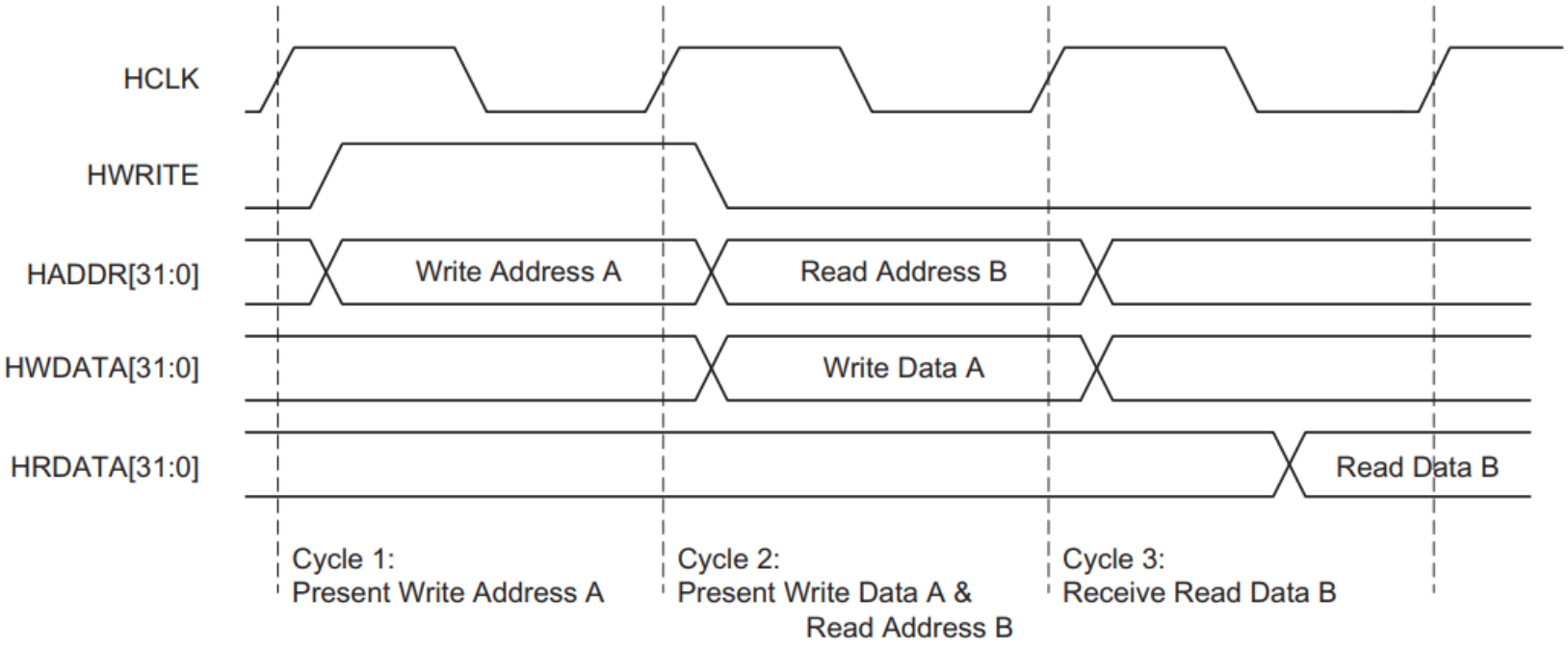


Figure e9.46 AHB-Lite transfer timing

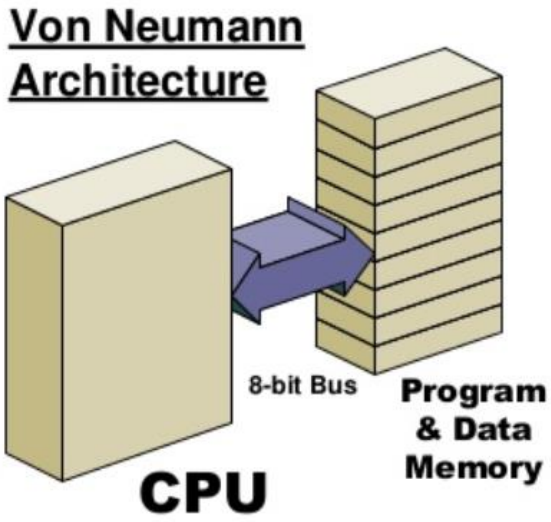
AHB

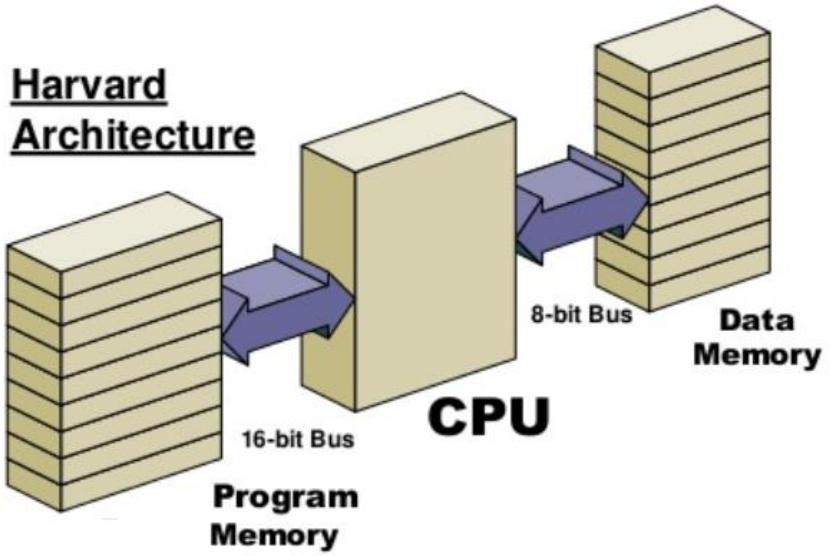
- GPIO uses AHB1 bus

Boundary address	Peripheral	Bus
0x5000 0000 - 0x5003 FFFF	USB OTG FS	AHB2
0x4002 6400 - 0x4002 67FF	DMA2	
0x4002 6000 - 0x4002 63FF	DMA1	
0x4002 3C00 - 0x4002 3FFF	Flash interface register	
0x4002 3800 - 0x4002 3BFF	RCC	
0x4002 3000 - 0x4002 33FF	CRC	
0x4002 1C00 - 0x4002 1FFF	GPIOH	AHB1
0x4002 1000 - 0x4002 13FF	GPIOE	
0x4002 0C00 - 0x4002 0FFF	GPIOD	
0x4002 0800 - 0x4002 0BFF	GPIOC	
0x4002 0400 - 0x4002 07FF	GPIOB	
0x4002 0000 - 0x4002 03FF	GPIOA	

The Computer Bus / Die Rekenaarbus

- Harvard and von Neumann architectures:
- A **von Neumann architecture** has only one bus which is used for both data transfers and instruction fetches, and therefore data transfers and instruction fetches must be scheduled - they can not be performed at the same time
- **Harvard architecture** has separate data and instruction busses, allowing transfers to be performed simultaneously on both busses.





- In Harvard architecture, a bridge is needed between data and instruction bus – if we programmatically want to change the program.
- Cache is used to speed up access to memory.

The System Bus / Die Stelselbus

- The ARM Cortex M4 is a Harvard Architecture.
- Three separate Advanced High-Performance Bus (AHB) interfaces
  - Instruction fetch from code memory (I-Code bus)
  - Data access to code memory (D-Code bus)
  - System bus – access to RAM (data memory) and peripherals (input/output)
- Private peripheral bus (PPB)

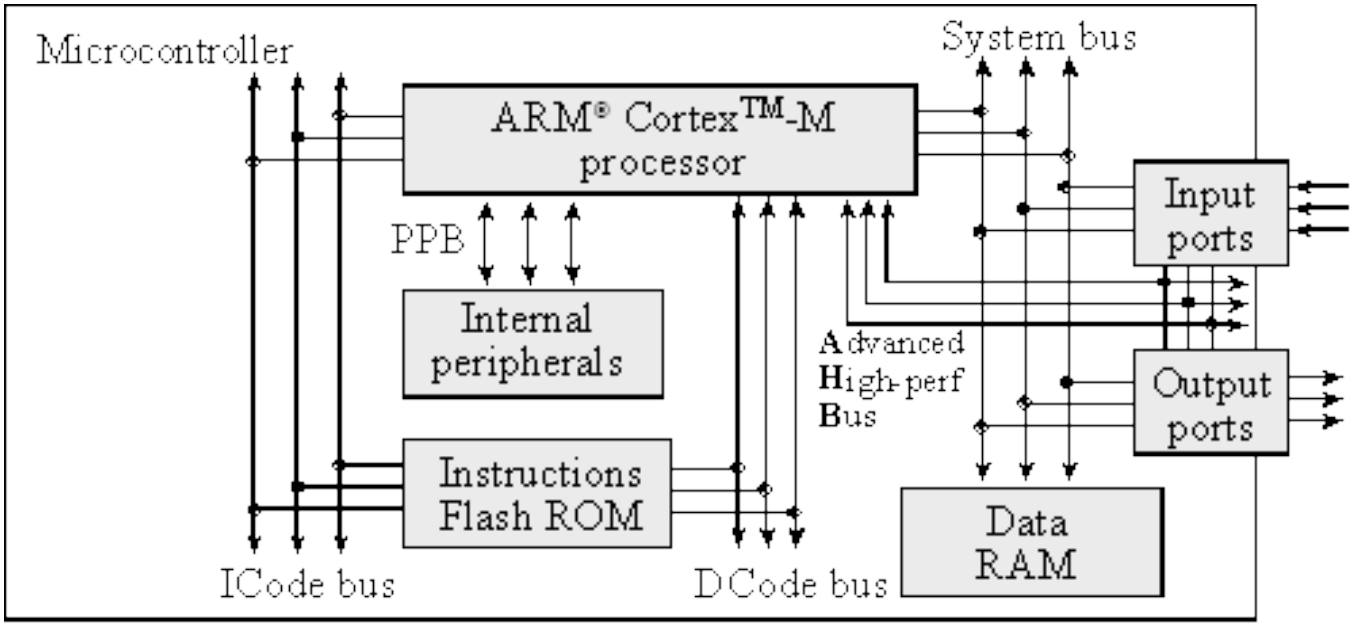
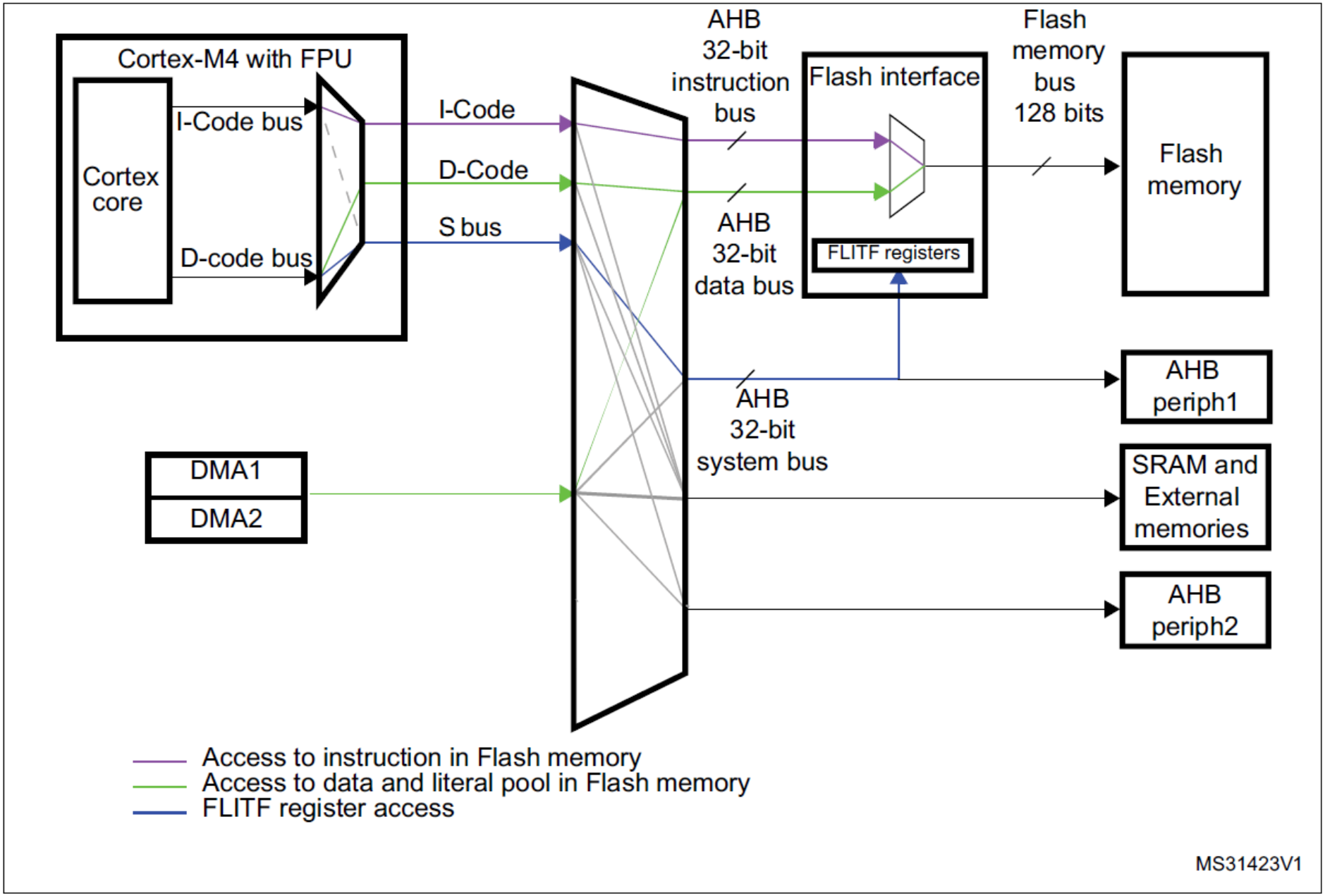
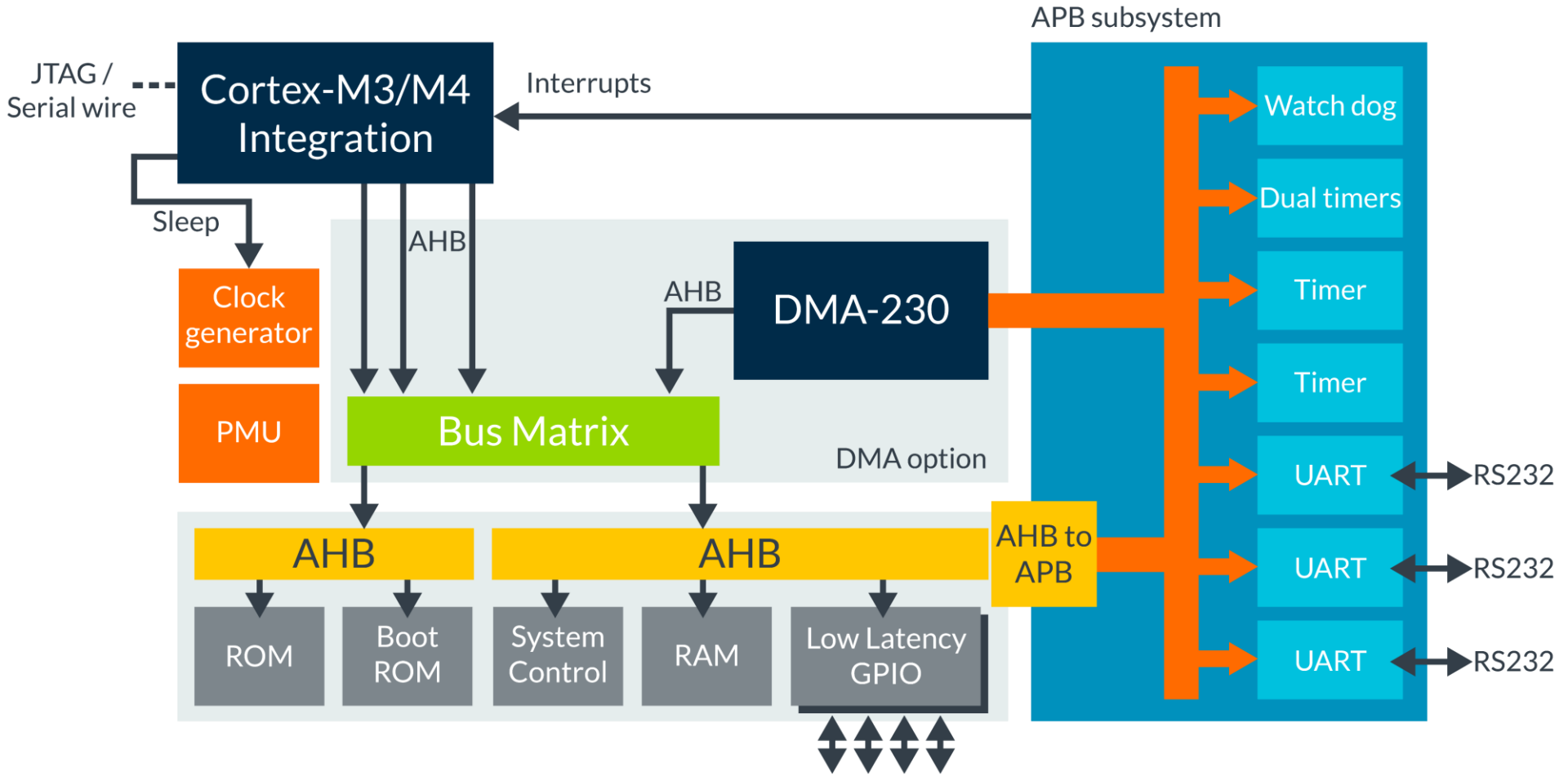


Figure 2. Flash memory interface connection inside system architecture (STM32F411xC/E)



The System Bus / Die Stelselbus



https://developer.arm.com/ip-products/subsystem/corstone-foundation-ip/cortex-m-system-design-kit

The System Bus / Die Stelselbus