Memory Systems

Memory Needs

Function	Description	Volatility	Size	Speed	Access	Portable	Technology
Data	Temp data/operand store for programs	V			Direct	N	DDR/SRAM
Instructions	Opcode storage for executing programs	V/NV			Direct	Y/N	DDR/SRAM/Flash
Parameters	BIOS settings, configurations	NV			Direct	N	EEPROM/Flash
OS/System	System and user programs	NV			Indirect	N	HD/SSD/SD/Flash
Data file store	Audio, video, other mass-store data	NV			Indirect	Y/N	HD/SSD/SD/Flash

Fast memory busses (like DDR) use different physical pin drivers than regular busses

Large-array memory busses (like HD) use high-speed serial busses like SATA or PCIe

Dynamic memory (like DDR) has much smaller cell size and much higher density, but must be refreshed

Main types of memory

Volatile (RAM): DDR, SRAM, PSRAM, Cellular RAM

Dynamic vs. Static	Dynamic: DDR, PSRAM, Cellular	Static: SRAM
High-speed vs. regular	High speed: > 200MHz	
Serial vs. parallel		
Multiplexed	Multiplexed port vs. Multiplexed part	
Synchronous/piped/burst	High throughput – device changes its own address	

Non-volatile (ROM): ROM, EEPROM, Flash, HD (magnetic)

Serial vs. parallel

In-system vs. Removable

Flash vs. EEPROM

Mass-store (magnetic and solid-state)

Micron: Leading supplier of RAM and ROM

Where's the SRAM?

Memory

DRAM
- DDR4 SDRAM
- DDR3 SDRAM
- DDR2 SDRAM
- DDR SDRAM
- SDRAM
- GDDR
- RLDRAM Memory
- LPDRAM

DRAM Modules
- RDIMM
- VLP RDIMM
- VLP UDIMM
- UDIMM
- SODIMM
- SORDIMM
- VLP Mini-DIMM
- LRDIMM
- Mini-DIMM
- NVDIMM

NAND Flash
- 3D NAND
- TLC NAND
- MLC NAND
- SLC NAND

Managed NAND
- e-MMC
- Embedded USB
- Universal Flash Storage

NOR Flash
- Parallel NOR Flash
- Serial NOR Flash
- Xccela™ Flash

Hybrid Memory Cube
- Short-Reach HMC

Multichip Packages
- UFS-Based MCP
- e.MMC-Based MCP
- NAND-Based MCP
- NOR-Based MCP

Micron: Leading supplier of RAM and ROM

Where's the SRAM?

Micron

128Mb: x8/x16, 3V, MT28EW Embedded Parallel NOR
Read AC Characteristics

Read AC Characteristics

Table 30: Read AC Characteristics – V_{CC}= V_{CCQ} = 2.7-3.6V

Parameter	Symbol		Condition	Min	Max	Unit	Notes
	Legacy	JEDEC					
Address valid to next address valid	^tRC	^tAVAV	CE# = V_{IL}, OE# = V_{IL}	70	–	ns	
Address valid to output valid	^tACC	^tAVQV	CE# = V_{IL}, OE# = V_{IL}	–	70	ns	
Address valid to output valid (page)	^tPAGE	$^tAVQV1$	CE# = V_{IL}, OE# = V_{IL}	–	20	ns	
CE# LOW to output valid	^tCE	^tELQV	OE# = V_{IL}	–	70	ns	
OE# LOW to output valid	^tOE	^tGLQV	CE# = V_{IL}	–	25	ns	
CE# HIGH to output High-Z	^tHZ	^tEHQZ	OE# = V_{IL}	–	20	ns	1
OE# HIGH to output High-Z	^tDF	^tGHQZ	CE# = V_{IL}	–	15	ns	1
CE# HIGH, OE# HIGH, or address transition to output transition	^tOH	^tEHQX, ^tGHQX, ^tAXQX	–	0	–	ns	
CE# LOW to BYTE# LOW	^tELFL	^tELBL	–	–	10	ns	
CE# LOW to BYTE# HIGH	^tELFH	^tELBH	–	–	10	ns	
BYTE# LOW to output valid	^tFLQV	^tBLQV	–	–	1	µs	
BYTE# HIGH to output valid	^tFHQV	^tBHQV	–	–	1	µs	

Memory

DRAM
- DDR4 SDRAM
- DDR3 SDRAM
- DDR2 SDRAM
- DDR SDRAM
- SDRAM
- GDDR
- RLDRAM Memory
- LPDRAM

DRAM Modules
- RDIMM
- VLP RDIMM
- VLP UDIMM
- UDIMM
- SODIMM
- SORDIMM
- VLP Mini-DIMM
- LRDIMM
- Mini-DIMM
- NVDIMM

NAND Flash
- 3D NAND
- TLC NAND
- MLC NAND
- SLC NAND

Managed NAND
- e-MMC
- Embedded USB
- Universal Flash Storage

NOR Flash
- Parallel NOR Flash
- Serial NOR Flash
- Xccela™ Flash

Hybrid Memory Cube
- Short-Reach HMC

Multichip Packages
- UFS-Based MCP
- e.MMC-Based MCP
- NAND-Based MCP
- NOR-Based MCP

ISSI: Leading supplier of non-leading-edge RAM and ROM

DRAM

- DDR4 SDRAM
- DDR3 SDRAM
- DDR3 SDRAM w/ ECC
- DDR2 SDRAM
- DDR SDRAM
- SDR SDRAM
- EDO & Fast Page Mode DRAM
- RLDRAM® 2/3

Mobile DRAM

- LPDDR2 SDRAM
- Mobile DDR SDRAM
- Mobile/Low Voltage SDR SDRAM

SRAM

- Asynchronous SRAM
- Serial SRAM & Low Pin Count SRAM
- Synchronous SRAM
- QUAD/QUADP & DDR-II/DDR-IIP
- CellularRAM/Pseudo SRAM
- HyperRAM™

MCP (Multi-Chip Package)

- LPDDR2 DRAM + Serial NOR Flash

Wafer Level Memory Solutions

- Wafer Level Memory Solutions

Analog

- Audio Amplifiers
- FxLED Driver
- Backlight Driver
- HBLED Driver
- I/O Expanders
- Sensor & Power Management
- Automotive Analog

Flash

- Serial NOR Flash
- Serial NOR Flash w/ECC
- Twin Serial NOR Flash
- HyperFlash™
- Parallel NOR Flash
- SPI NAND Flash
- NAND Flash
- eMMC
- Programmer Support
- Flash Application Notes

ISSI: Leading supplier of non-leading-edge RAM and ROM

IS61WV102416FALL
IS61/64WV102416FBLL

AC CHARACTERISTICS (OVER OPERATING RANGE)

READ CYCLE AC CHARACTERISTICS

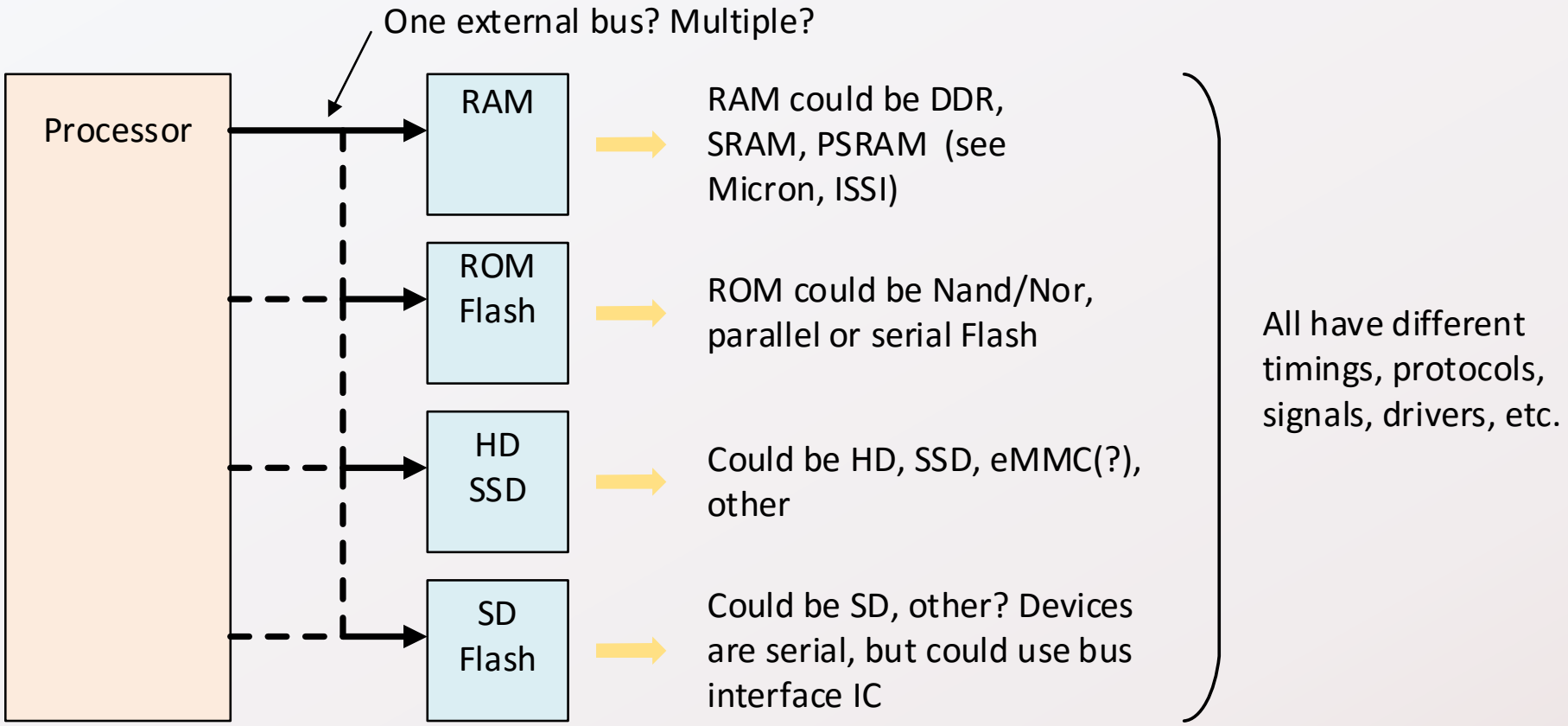
Parameter	Symbol	-8(1)		-10(1)		-20(1)		unit	notes
		Min	Min	Min	Max	Min	Max		
Read Cycle Time	tRC	8	-	10	-	20	-	ns	
Address Access Time	tAA	-	8	-	10	-	20	ns	
Output Hold Time	tOHA	2.5	-	2.5	-	2.5	-	ns	
CS# Access Time	tACE	-	8	-	10	-	20	ns	
OE# Access Time	tDOE	-	5.5	-	6	-	8	ns	
OE# to High-Z Output	tHZOE	0	4	0	5	0	8	ns	2
OE# to Low-Z Output	tLZOE	0	-	0	-	0	-	ns	2
CS# to High-Z Output	tHZCE	0	4	0	5	0	8	ns	2
CS# to Low-Z Output	tLZCE	3	-	3	-	3	-	ns	2
UB#, LB# Access Time	tBA	-	5.5	-	6	-	8	ns	
UB#, LB# to High-Z Output	tHZB	0	4	0	5	0	8	ns	2
UB#, LB# to Low-Z Output	tLZB	0	-	0	-	0	-	ns	2

Bus partitions

Typically one physical bus, but many different potential devices/technologies.

Different devices have different timing, control, drive levels, etc

Different address ranges on same bus may need to behave differently



External Memory Interface (Busses)

Should processor bus be designed for:

DDR? SRAM? What speed? What width?

Can bus accommodate different devices or technologies?

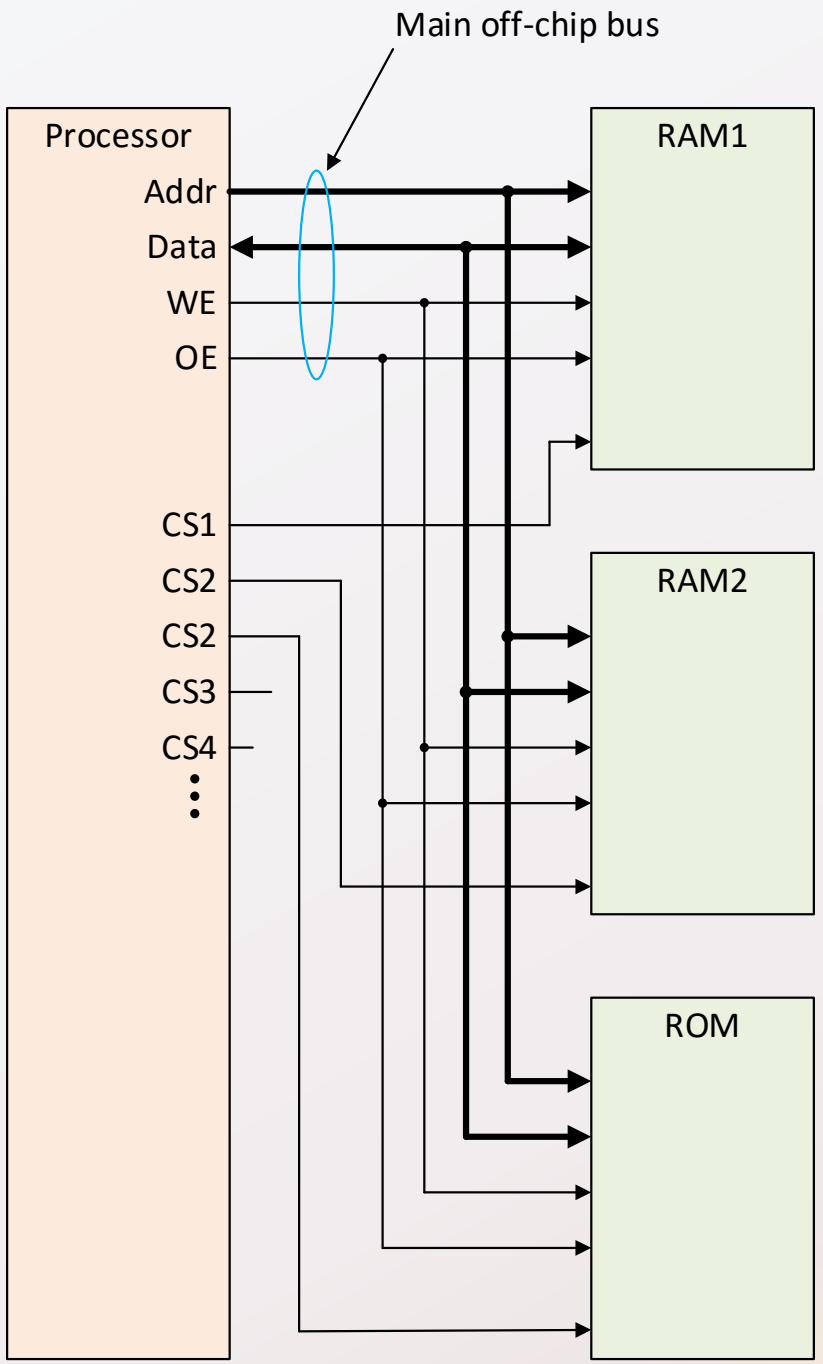
Different access speeds?

Bus Grant/Bus acknowledge

Bus hold-off (dynamic wait state)

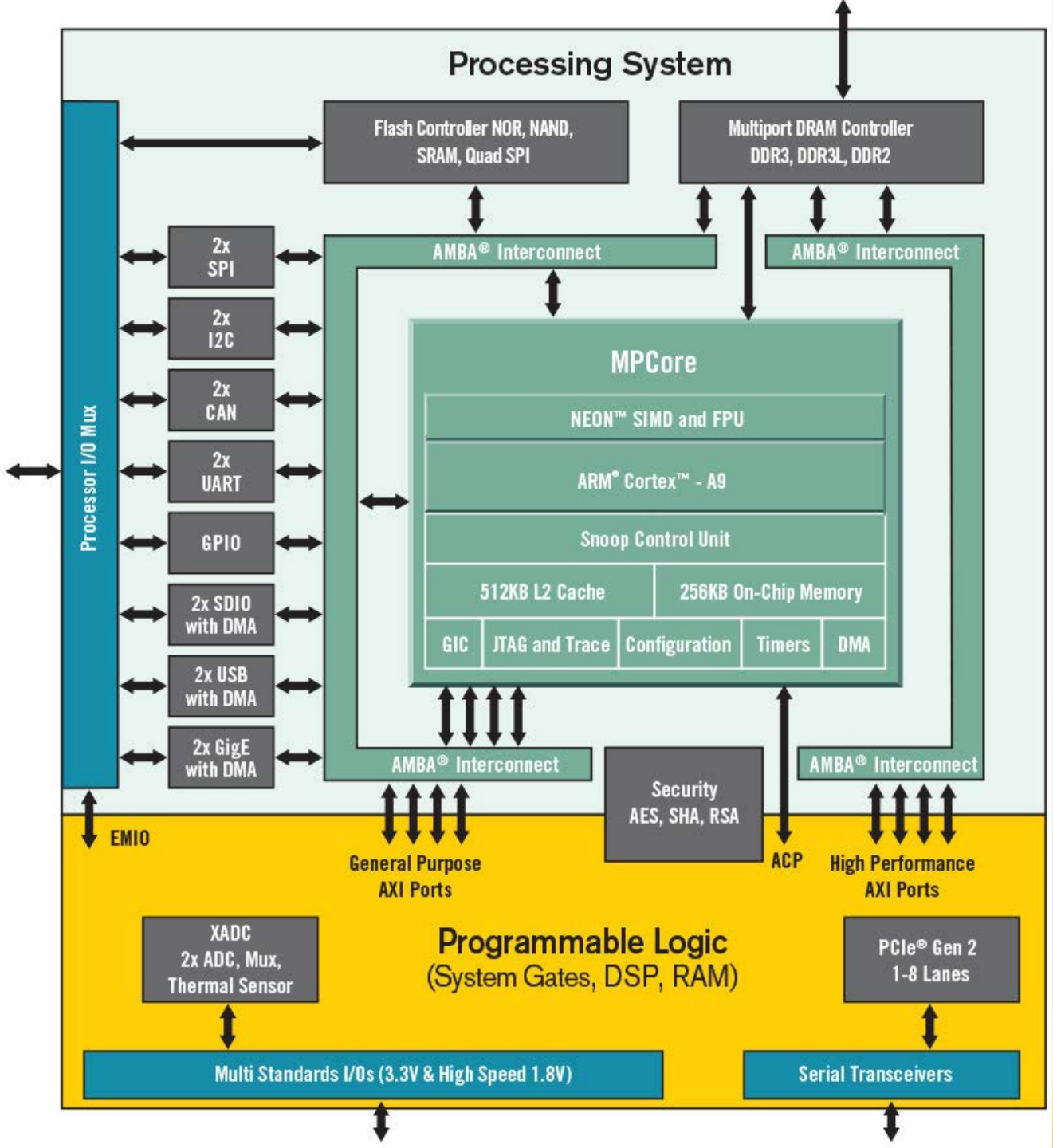
Different control signals?

Different signal drive/protocols?



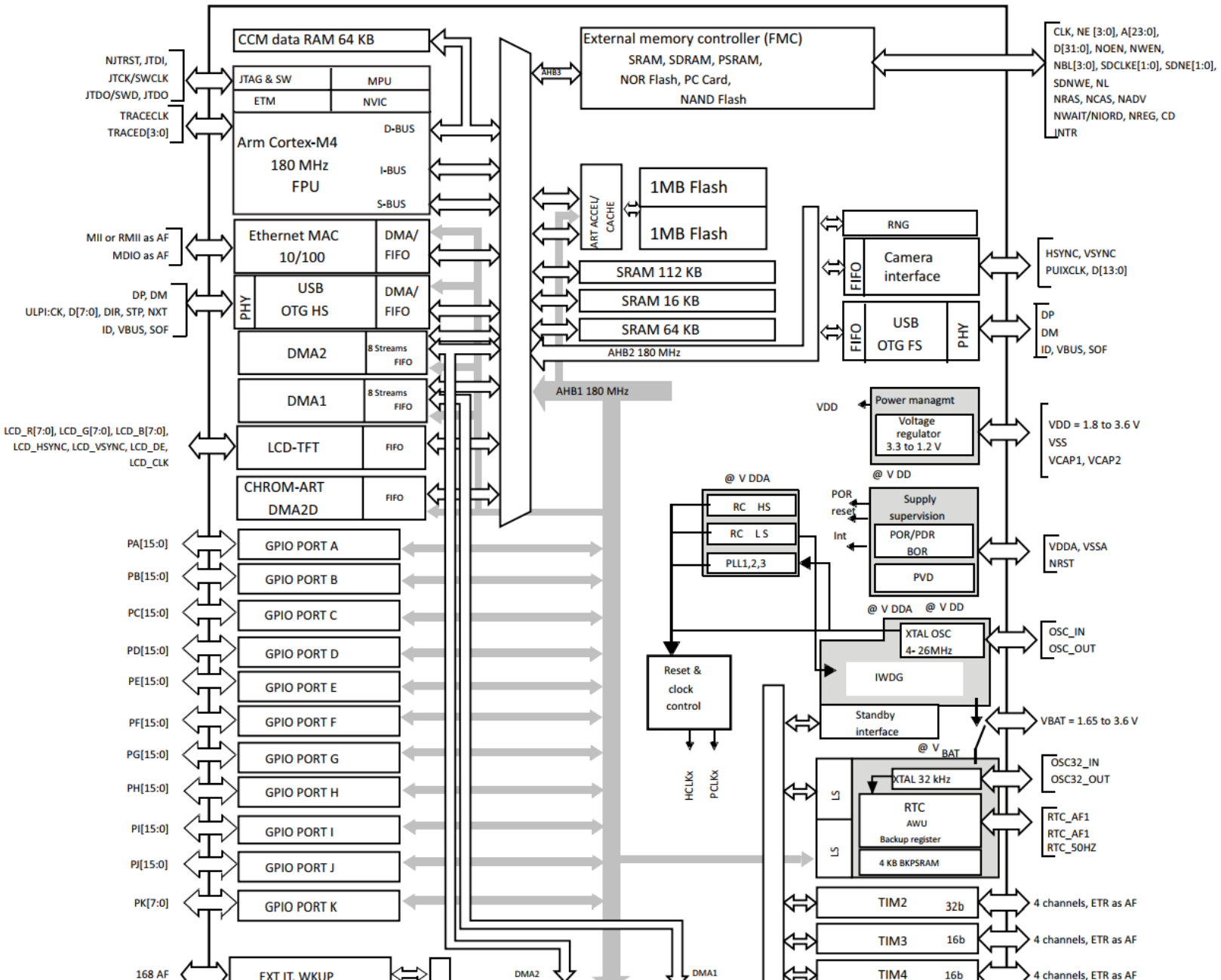
ZYNQ

External busses typically target one type of memory; if more flexibility needed, more busses used



ST ARM

Figure 4. STM32F427xx and STM32F429xx block diagram



PIC32

32-bit PIC® and SAM Microcontrollers Peripheral Integration

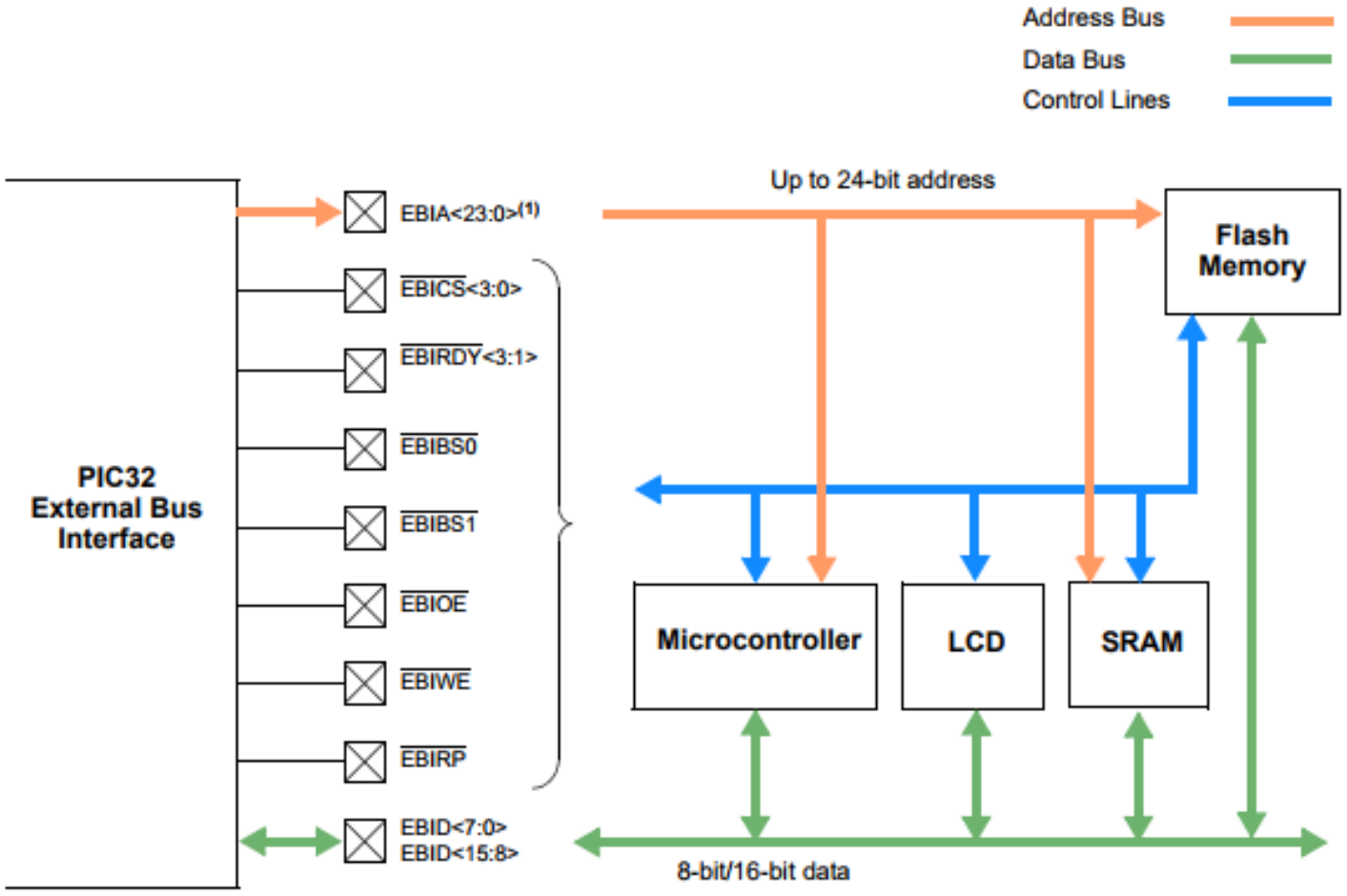
Quick Reference Guide

http://ww1.microchip.com/downloads/en/DeviceDoc/60001455C.pdf

Product Family	Core	Max. Operating Frequency (MHz)	Program Flash Memory (KB)	RAM (KB)	Pin Count	Peripheral Function Focus: Intelligent Analog				Waveform Control			Timing and Measurements			Safety and Monitoring		Communications										User Interface			Security		System Flexibility						
						ADC (channels/bits)	ADC Speed (sps)	DAC (channels/bits)	Analog Comp. (+Op Amp)	Output Compare Channels	Input Capture Channels	PWM Channels	16-bit/32-bit Timer	TCC (24-bit Control Timer)(4)	Motor Interface (QEI/QDEC)(4)	Watchdog Timer Dead Man Timer (DMT)(4)	Class B Safety/DSU/Touch Safety	USB (FS/HS) + PHY (Trx)	CAN (2.0B or FD)	Ethernet (10/100)	SERCOM/FLEXCOM(4)	USART/UART	I²C	SPI(1)	SDIO/SD/eMMC	CMOS Camera Interface	SQI/QSPI	Audio CODEC (I²S)(4)	Peripheral Bus Interface PMP/EBI (Bus width, bit)(4)	Touch (PTC/CTMU, channels)(4)	Segment/Graphics LCD Controller	LCD/GFX Interface (PMP/EBI)	Crypto Engine (AES, SHA, ECC, RSA/DSA, TRNG)	Tamper Detection	Dual Panel/Bank Flash(4)	Intelligent Low Power Peripheral Event System (channels)(4)	DMA (channels)	Low Active Power (µA/MHz)	5V Support	CLC/CCL(4)	Ultra-Small Package (WLCSP)
PIC32 Family																																							
PIC32MM	microAptiv™	25	16–64	4–8	20–36	14/12	200k	1/5	2	3	3	8	7/3			W						2		2				2										✓	
PIC32MX1/2*/5*+	M4K	50	16–512	4–64	28–100	48/10	1M		3	5	5	5	5/2			W	B	1F+P*	1+			5	2	4				4	P16	C13		P					4				
PIC32MX1/2 XLP	M4K	72	128–256	32–64	28–44	13/10	1M		3	5	5	5	5/2			W+D	B	1F+P				2	2	2				2	P12	C13		P					4	✓			
PIC32MX3/4*	M4K	120	32–512	16–128	64–124	16/10	1M		2	5	5	5	5/2			W	B	1F+P*				5	2	2				2	P16	C13		P					4				
PIC32MX5	M4K	80	64–512	16–64	64–100	16/10	1M		2	5	5	5	5/2			W	B	1F+P	1			6	5	4					P16			P					8				
PIC32MX6	M4K	80	64–512	32–128	64–100	16/10	1M		2	5	5	5	5/2			W	B	1H+P		1		6	5	4					P16			P					8				
PIC32MX7	M4K	80	128–512	32–128	64–100	16/10	1M		2	5	5	5	5/2			W	B	1F+P	2	1		6	5	4					P16			P					8				
PIC32MKGP/MC	microAptiv	120	512–1024	128–256	64–100	42/12	16M	3/12	5O4	12	16	16	14/16		E	W+D	B	2F+P	4			6		6				6	P24			P			✓		13				
PIC32MZEF(3)	M-Class	252	512–2048	128–512	64–144	48/12	18M		2	9	9	9	9/4			W+D	B	1H+P	2	1		6	5	6			✓	6	P/E24			P+E	A,S,T		✓		18				
PIC32MZDA(2)	microAptiv	200	1024–2048	256–640	169–288	45/12	18M		2	9	9	9	9/4			W+D	B	1H+P	2	1		6	5	6	1		✓	6	P/E24	C35	G	P+E	A,S,T		✓		26				
SAM Family																																							
SAMD09	CM0+	48	8–16	4	14–24	10/12	350k	1/10	2	6	3	4	2/1			W	B				**2**	2	2	2												6	6	✓			
SAMD10	CM0+	48	8–16	4	14–20	10/12	350k	1/10	2	6	3	12	2/1	1		W	B+T				**3**	3	3	3						P72						6	6	✓			✓
SAMD11	CM0+	48	16	4	14–24	10/12	350k	1/10	2	6	3	12	2/1	1		W	B+T	1F+P			**3**	3	3	3						P72						6	6	✓			✓
SAMD20	CM0+	48	16–256	2–32	32–64	20/12	350k	1/10	2	16	8	16	5/2			W	B+T				**6**	6	6	6						P256						8		✓			✓
SAMD21	CM0+	48	32–256	4–32	14–64	20/12	350k	1/10	2	18	8	24	5/2	3		W	B+T	1F+P			**6**	6	6	6				1		P256						12	12	✓			✓
SAMD21L	CM0+	48	32–64	4–8	32–48	18/12	350k		4	18	13	24	5/2	3		W	B+T				**5**	5	5	5												12	12	✓			
SAMDA1(3)	CM0+	48	16–64	4–8	32–64	20/12	350k	1/10	2	18	8	24	5/2	3		W	B+T	1F+P			**6**	6	6	6				1		P256						12	8	✓			
SAML21	CM0+	48	32–256	4–32	32–64	20/12	1M	2/12	2O3	24	8	24	5/2	2		W	B+T	1F+P			**6**	6	6	6						P169			A,T			12	16	✓VBAT		✓	✓
SAML22	CM0+	32	64–256	8–32	48–100	20/12	1M		2	12	8	12	4/2	1		W	B+T	1F+P			**6**	6	6	6						P256	S320		A,T	✓		8	16	✓VBAT		✓	✓
SAMC20	CM0+	48	32–256	4/32	32–64	12/12	1M		2	14	6	18	5/2	2		W	B+T				**4**	4	4	4						P256						6	6	✓	✓	✓	✓
SAMC21(5)	CM0+	48	32–256	4–32	32–100	20/12	1M	1/10	4	18	8	24	5/2	2		W	B+T		2FD		**8**	8	8	8						P256						12	12	✓	✓	✓	✓
SAM4N	CM4	100	512–1024	64–80	48–100	16/10	510k	1/10		18	12	4	2/–		D	W						3/4	3	4													23	✓			
SAM4S	CM4	120	128–2048	64–160	48–100	16/12	1M	2/12	1	18	12	4	2/–		D	W		1F+P				2/2	2	3	1	✓		1	E24			E			✓	14	22	✓			✓
SAM4E	CM4F	120	512–1024	128	100–144	24/12	300k	2/12	1	24	18	4	–/3		D	W		1F+P	2	1		2/2	2	3	1	✓			E24			E	A	✓			33	✓			
SAM4L	CM4	48	128–512	32–64	48–100	16/12	300k	1/10	4	18	12	5	2/–			W		1F+P				4/1	4	5		✓		1		P32	S160		A,T			4	16	✓			✓
SAMG	CM4F	120	256–512	64–176	49–100	8/12	500k			6	6	6	2/–			W		1F+P			**8**	8	8	8				2						✓		6	30	✓			✓
SAMD5x	CM4F	120	256-1024	128-256	64-128	32/12	1M	2/12	2	25	16	24	8/4	2	D	W	B+T	1F+P			**8**	8	8	8	2	✓	✓	1		P32			A,S,E,R,T	✓	✓	32	32	✓		✓	✓
SAME5x	CM4F	120	256-1024	128-256	64-128	32/12	1M	2/12	2	25	16	24	8/4	2	D	W	B+T	1F+P	2FD	1	**8**	8	8	8	2	✓	✓	1		P32			A,S,E,R,T	✓	✓	32	32	✓		✓	
SAMS7x(2)	CM7	300	512–2048	256–384	64–144	24/12	2M	2/12	1	44	24	8	4/–		D	W		1H+P				3/5	3	5	1	✓	✓	2	E24			E	A,S,T	✓		12	24	✓			
SAME7x(2)	CM7	300	512–2048	256–384	64–144	24/12	2M	2/12	1	44	24	8	4/–		D	W		1H+P	2FD	1		3/5	3	5	1	✓	✓	2	E24			E	A,S,T	✓		12	24	✓			
SAMV7x(2)(3)	CM7	300	512–2048	256–384	64–144	24/12	2M	2/12	1	44	24	8	4/–		D	W		1H+P	2FD	1		3/5	3	5	1	✓	✓	2	E24			E	A,S,T	✓		12	24	✓			

1: USARTs with SPI mode are taken into account **2:** DRAM Memory Support: PIC32MZ DA with DDR2 (32 MB embedded or 128 MB external); SAMS7x/E7x/V7x with SDRAM (external) **3:** Automotive-grade devices **4:** Terminology on back
5: SAMC20/C21 are true 5V devices; SAMC21 also comes with 3× 16-bit Delta-Sigma ADC (*: Variants with USB function) (+: Variants with CAN function)

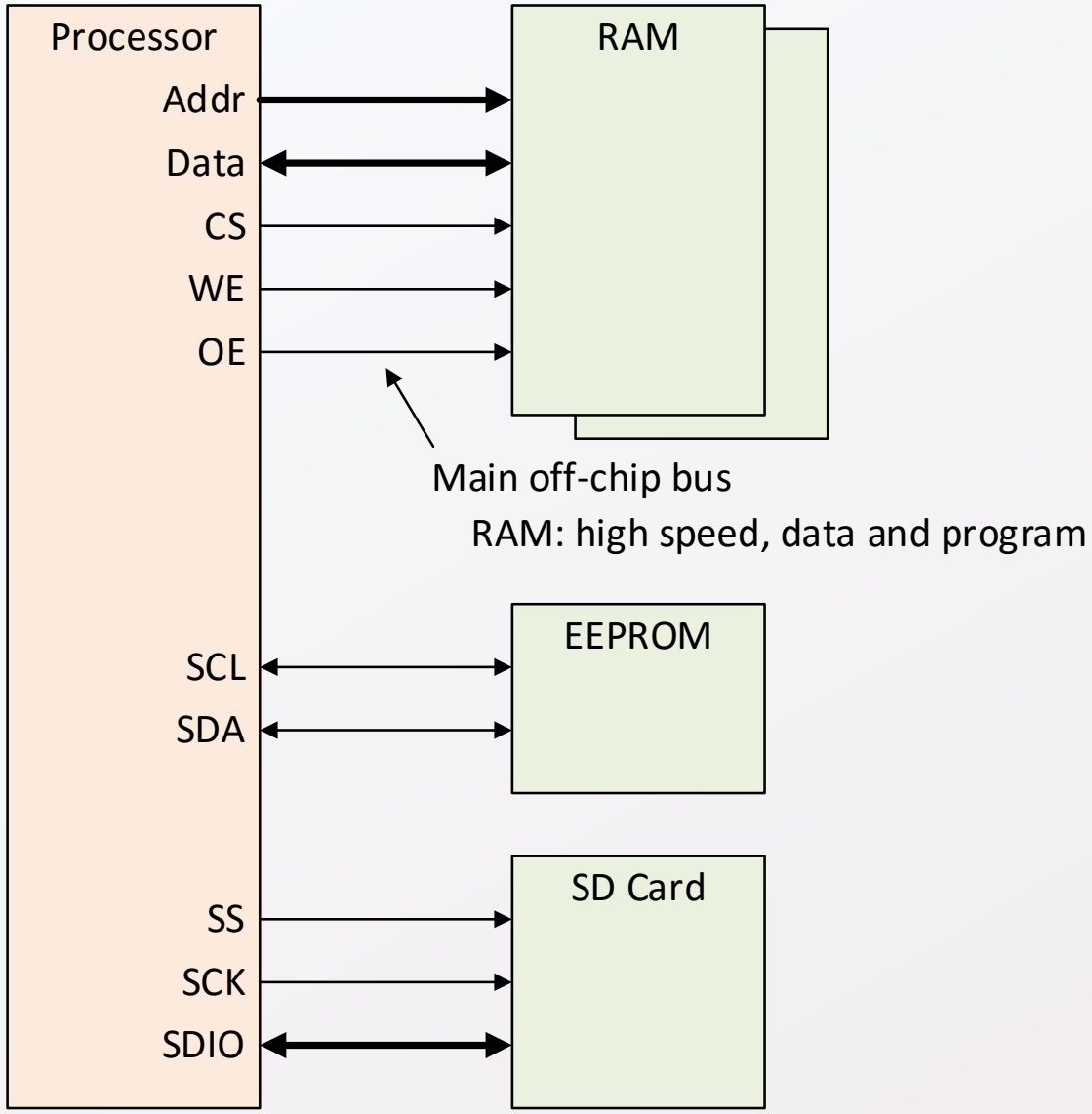
Figure 47-2: EBI Module Pinout and Connections to External Devices



Note 1: No EBIA address pins are available on 64-pin devices and the EBIA<23:20> address pins are only available on 144-pin devices.

http://ww1.microchip.com/downloads/en/DeviceDoc/60001245A.pdf

A typical higher-performance configuration



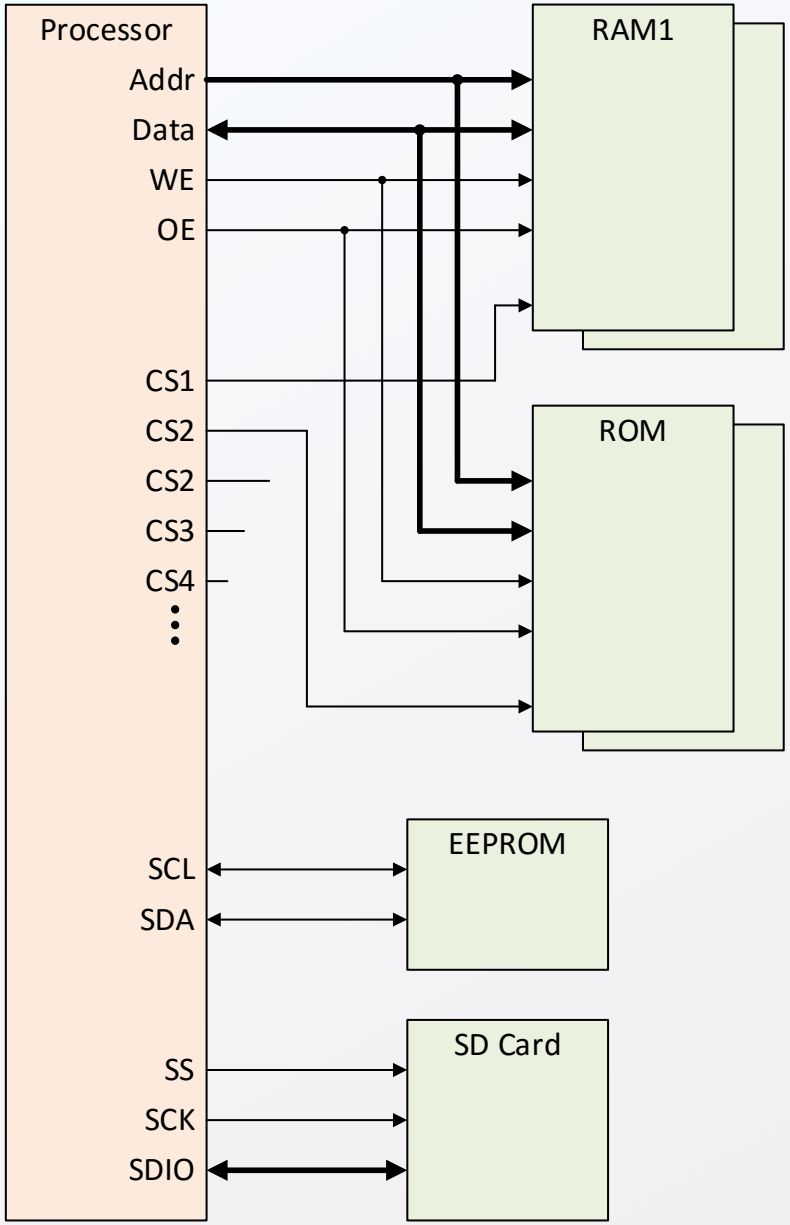
Single bank of homogenous RAM

If multiple chips, all are similar

System software loaded from ROM into RAM for execution

ROM may be HD/SSD, SD, or network

A typical lower-to-midrange-performance configuration

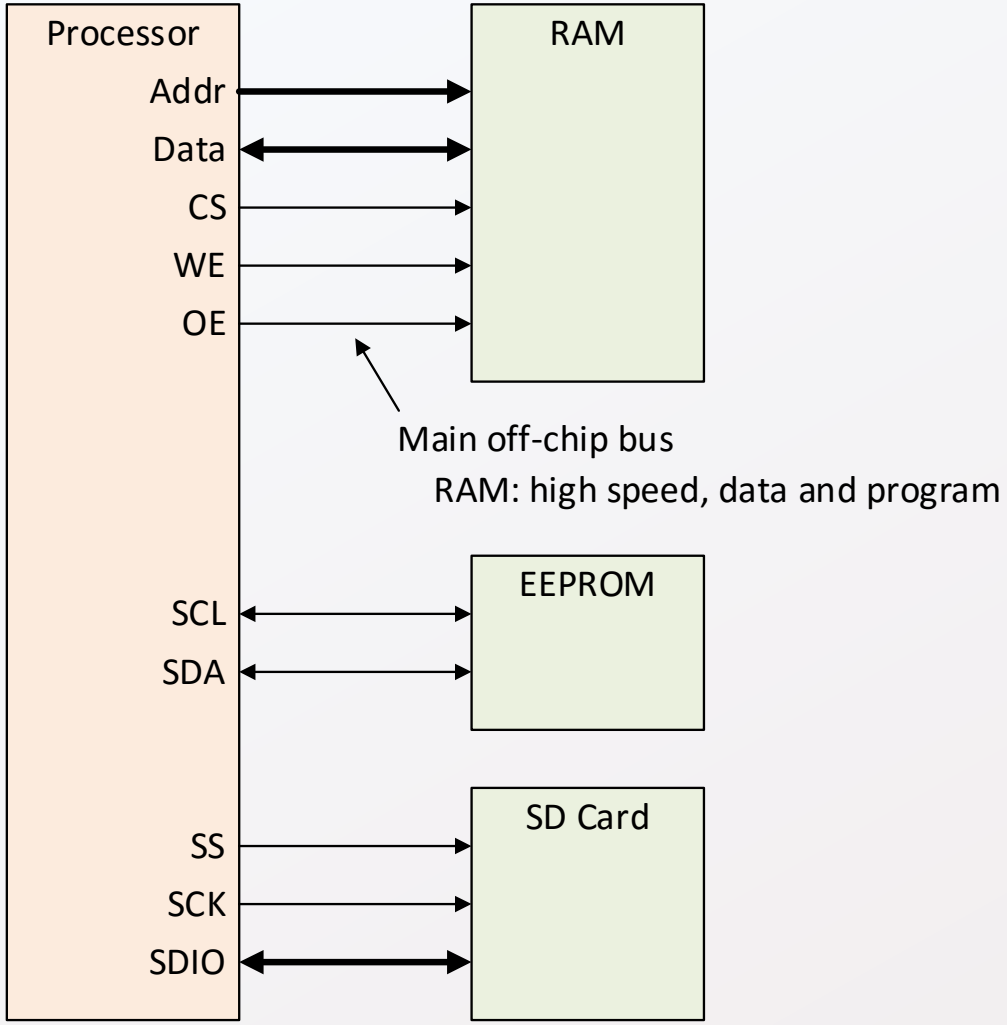


Single bank of homogenous RAM

If multiple chips, all are similar

Single bank of parallel ROM that shares same bus; timings similar enough

Booting



Ram is loaded from external source (like SD card) during “boot” sequence

(Some processors execute from parallel ROM)

What is a “boot” sequence?

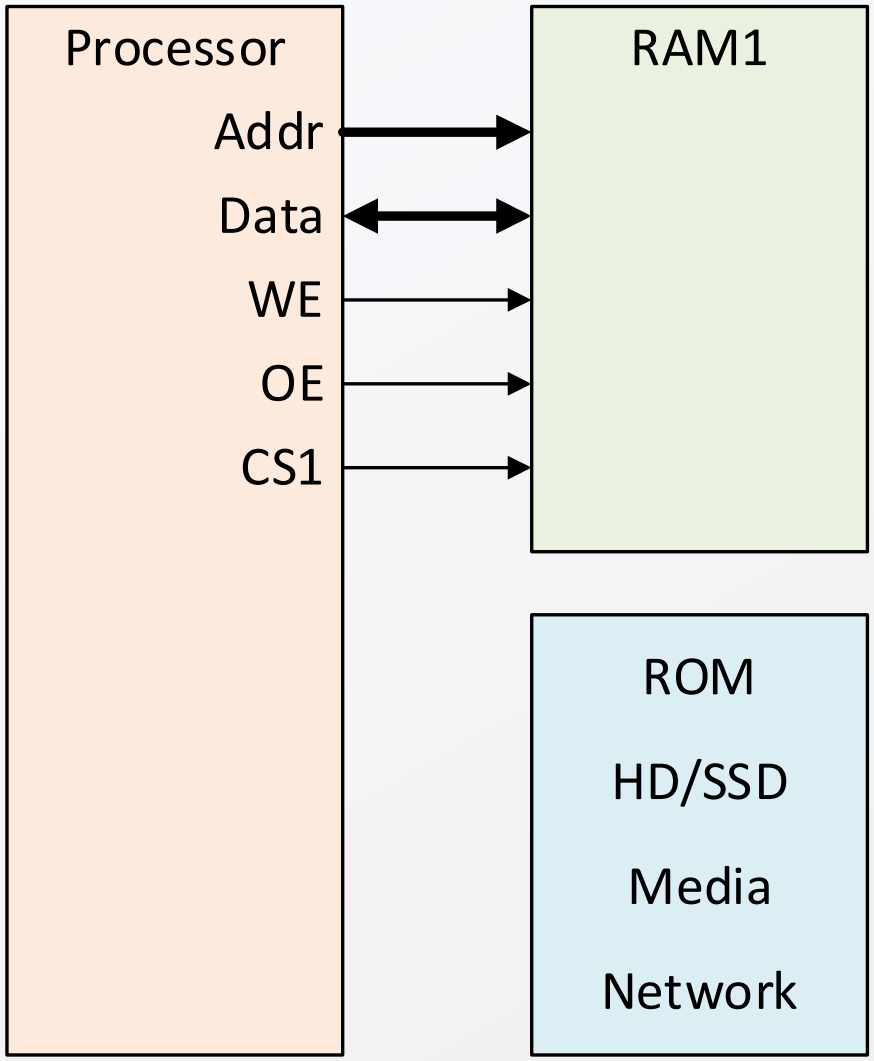
Pull oneself up by your own bootstraps – i.e., improve the situation without outside help, in small measured steps

Power applied, reset released, then what?

Fetch first instruction… from where?

Fixed address; must contain ROM that responds to “natural bus cycle”

Program Execution



Source code you develop is stored in a file managed by the operating system. The tools compile the source, create an .elf (or other format) object file that can be placed in memory and directly executed.

The .elf was built using a symbol table that contains relative addresses for data and program redirects. It can be loaded anywhere in RAM for execution.

Where should it be placed? How can it be placed?

Should the source file creator need to know where it might end up?

Models

One programmer, one application: program can be “hard located” in memory

One programmer, multiple applications: programs may be “hard located” in memory

Multiple programmers: programs can’t be “hard located” in memory – how could they be, given changing code size, changing stack/heap size, changing data needs

How to inform programmers where to locate programs? Can’t!

Solution: Every program gets the same view of memory and resources – every program can assume it is the only program.

Implementation: Does the OS sort out what’s loaded where? Would that even work?

- Absolute vs. relative references in code base

Virtual Memory

Only practicable solution: hardware does dynamic address translation in real-time

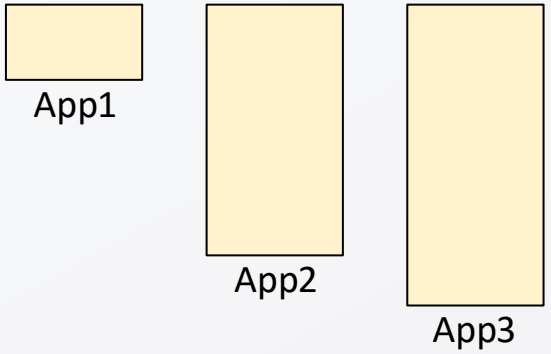
- Every single LOAD/STORE memory address passes through MMU and gets translated
- All programs can be written as if they were the only program running
- OS assigns physical memory location
- MMU maps every address to a physical address on the fly

Compiler/Assembler do not need to know about execution environment

- Note: PC-relative indexing works regardless of physical addresses

Virtual Memory

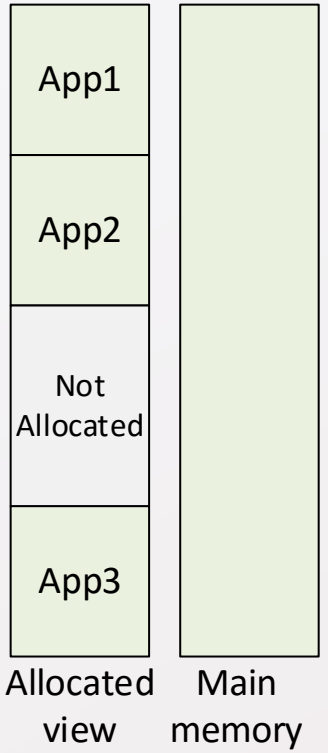
OS allocates memory to applications. Allocated memory need not be same size as executable code.



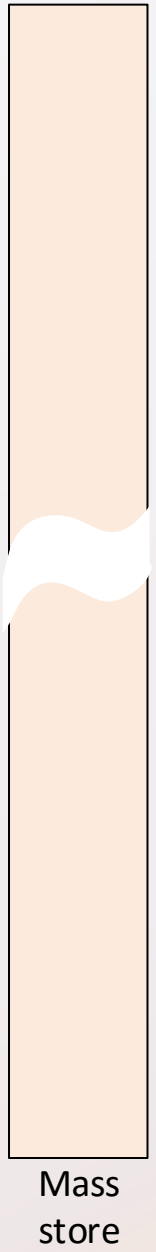
All application programs assume they start at address 0, and they rely on the OS to assign a location in physical memory, and the MMU to translate virtual addresses in real time.

Some applications are larger than their allocated memory.

More applications may be pending; some applications may need to surrender their main memory allocations to allow other programs to run.



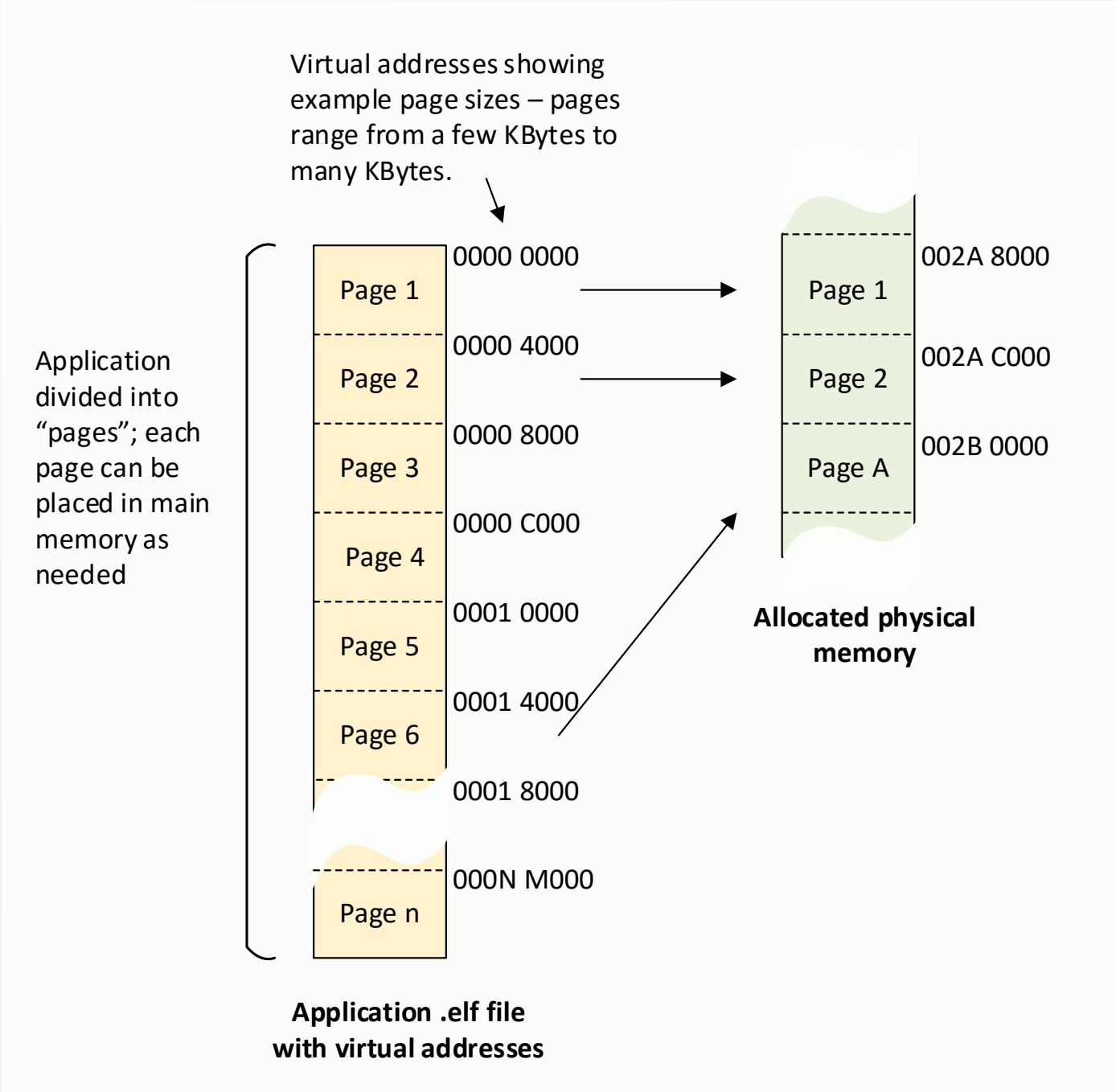
Main memory is much smaller and much faster than mass-store memory where non-executing programs are stored.



Virtual Memory

Files are divided into “pages”. Pages are swapped in and out of physical memory as needed.

If an access to a non-resident page is requested, a “page fault” occurs and processing must stop until page is loaded.



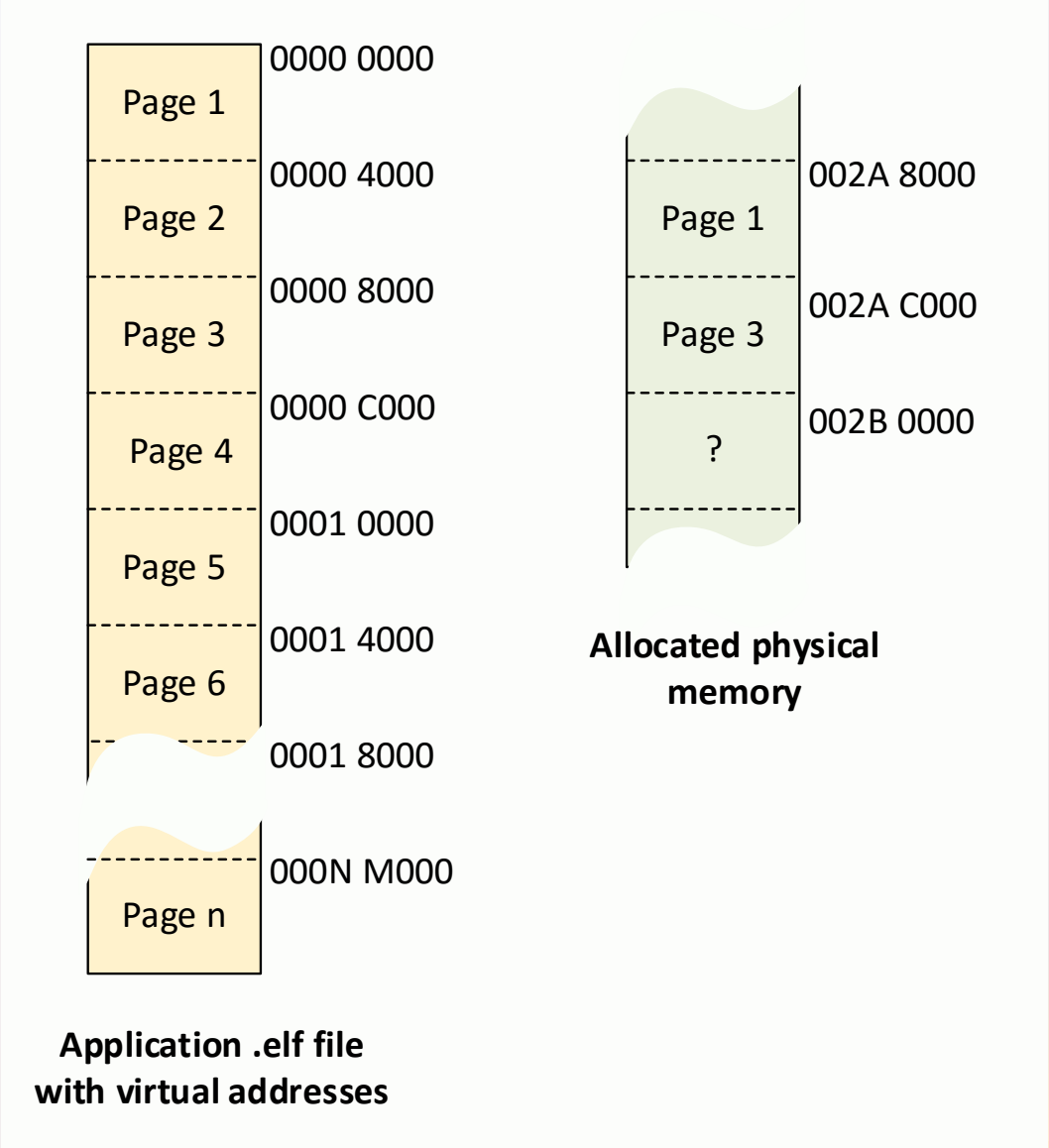
Virtual Memory: Constructing physical addresses

This example uses a 16K page; the lower 14 bits are the same for the virtual and physical address.

For a 32-bit address, the upper 18 bits must be “mapped” from the virtual address into the correct physical address space.

A “page table” stored in main memory stores the upper 18 bits of the physical base address of each page. The address stored for page 3 is 002AC.

Virtual address 0000 8123 would map to physical address 002A C123.



Virtual Memory: Translation look-aside buffer

If every translation required an access to main memory to retrieve the upper 16 bits, the memory would run at half speed. Some number of page table entries are stored in a cache called the TLB.

If the referenced page is in physical memory, it's "tag" (the upper 16 bits) will be in the TLB and the physical address can be constructed quickly.

How is the tag located? CAM.