

Computer Memory

Textbook: Chapter 1

ARM Cortex-M4 User Guide (Section 2.2 – Memory Model)

STM32F4xx Technical Reference Manual:

Chapter 2 – Memory and Bus Architecture

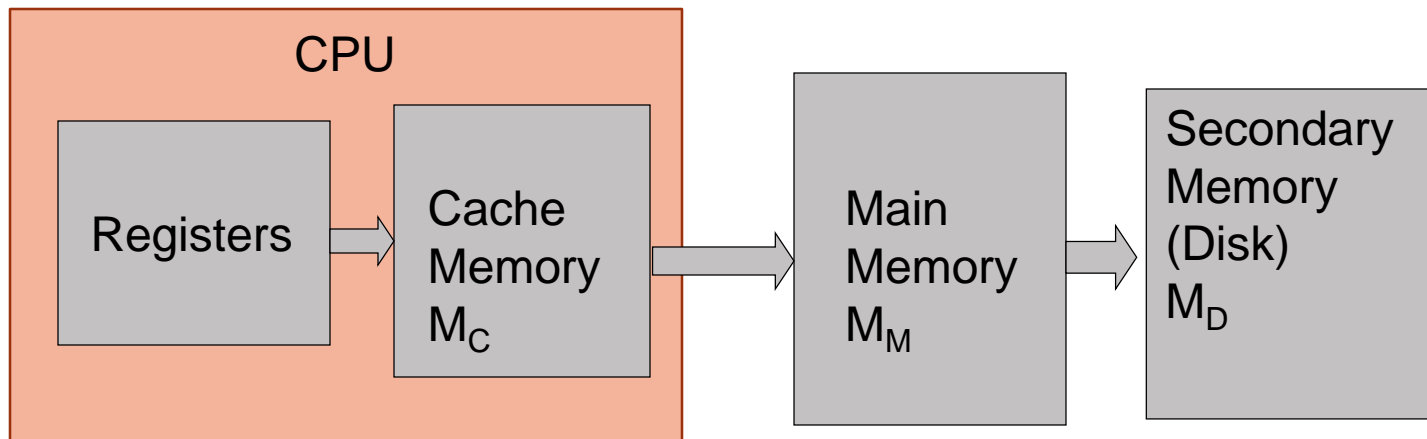
Chapter 3 – Flash Memory

Chapter 36 – Flexible Static Memory Controller

Computer Memory Systems

- Memory system hierarchy
 - Disk, ROM, RAM, Cache
- Memory module (chip) organization
 - On-chip (address) decoder, cell array
- Memory system interfacing
 - Address decoding
 - Bus timing
- Direct memory access (DMA)
 - Transfer data directly between memory and I/O devices
 - Coordinated by a DMA controller

Computer Memory Hierarchy



Memory Content: $M_C \subseteq M_M \subseteq M_D$

Memory Parameters:

- Access Time: increase with distance from CPU
- Cost/Bit: decrease with distance from CPU
- Capacity: increase with distance from CPU

Semiconductor Memory

- RAM (Random Access Memory)
 - Constant access time, independent of location
 - A unique address for each location (generally a byte)
 - The address is decoded by one or more address decoders
- RAM (Read/Write Memory) vs. ROM (Read Only Memory)
 - RAM
 - User's application programs and data
 - Information is lost when the power is off
 - ROM
 - Embedded system program code and operating system
 - Information is retained even without power
 - Each ROM cell is simpler than a RAM cell

Read-only memory types

- Mask-programmed ROM
 - Programmed at factory
- PROM (Programmable ROM)
 - Programmable once by users
 - Electric pulses selectively applied to “fuses”
- EPROM (Erasable PROM)
 - Repeatedly programmable/reprogrammable
 - Electric pulses for programming (seconds)
 - Ultraviolet light for erasing (minutes)
- EEPROM (Electrically Erasable PROM)
 - Electrically erasable at the single-byte level (*msec*) & programmable
- Flash EPROM
 - Electrically programmable (*µsec*) & erasable (block-by-block: *msec*~*sec*)
 - Most common program memory in embedded applications
 - Widely used in digital cameras, multimedia players, smart phones, etc.

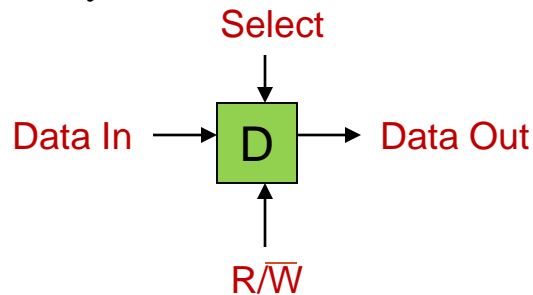
ROM devices are “non-volatile” – they retain information, even when not powered.

Read-write memory types

- Static RAM (SRAM)
 - Each cell is a flip-flop, storing 1-bit information
 - Information is retained as long as power is on (lost when power off)
 - Faster than DRAM
 - Requires a larger area per cell (more transistors) than DRAM
- Dynamic RAM (DRAM)
 - Each cell is a capacitor, which needs to be refreshed periodically to retain the 1-bit information
 - A refresh consists of reading followed by writing back
 - Refresh overhead

Memory organization (RAM)

- RAM Structure
 - Memory Cell



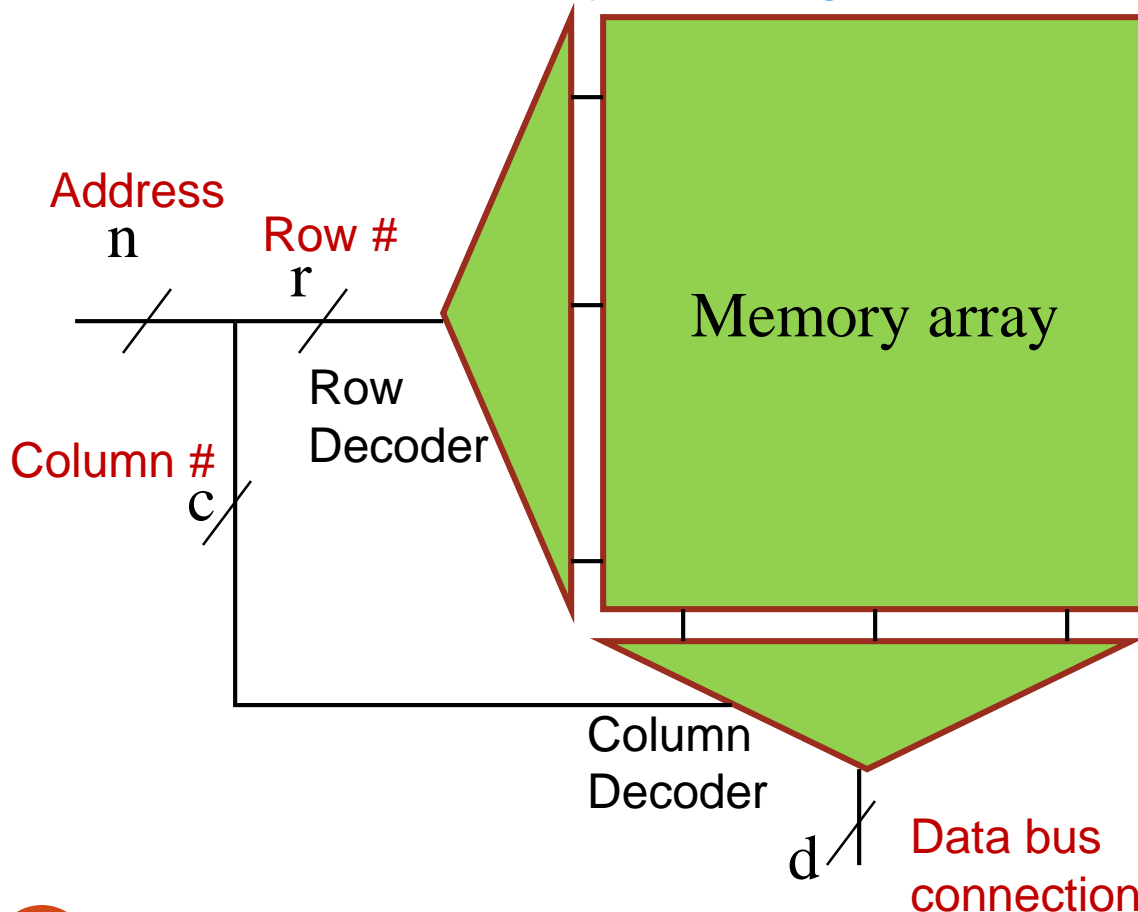
- A *byte* consists of 8 memory cells, with common control signals, *Select* and $\overline{R/W}$, and 8 bidirectional data lines.

Some RAMs have separate Din and Dout

- With n -bit address, the memory system can contain up to 2^n bytes.
An n -bit address is decoded by one or more address decoders to generate the control signal, *Select*.

ROM/RAM device organization

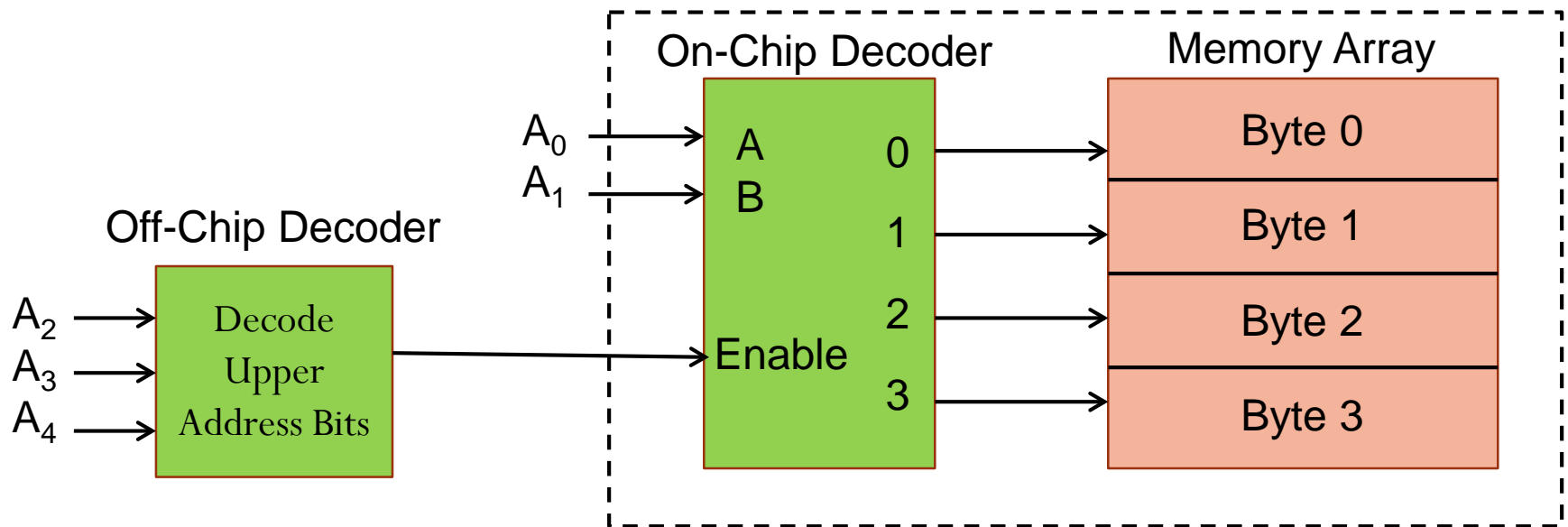
Memory "organization" = $2^n \times d$
(from system designer's perspective)



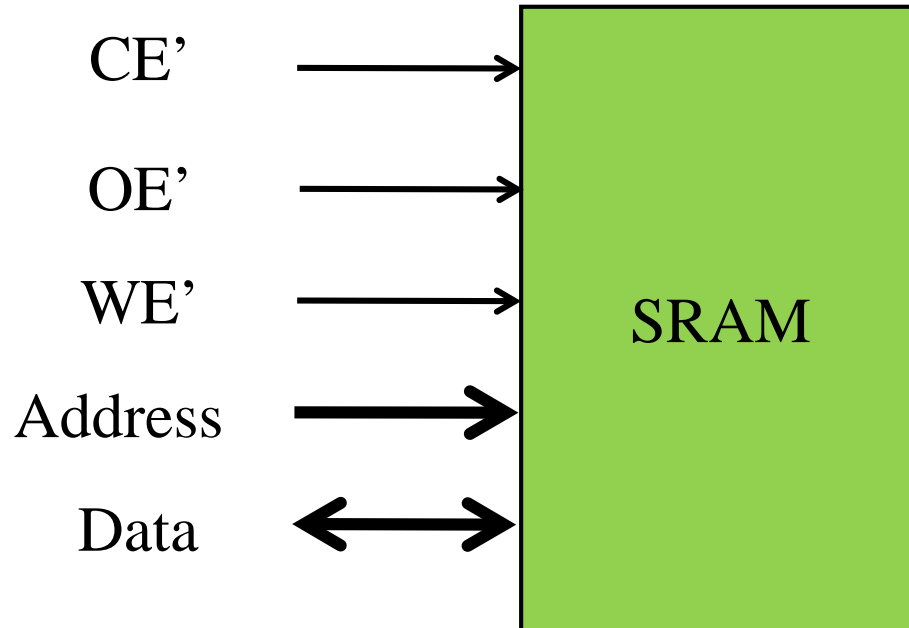
- Size.
 - 2^n addressable words
 - Address width =
 $n = r + c$
- Aspect ratio.
 - Data width d .

Address Decoding

- Selecting a sub-space of memory address
 - A simple example
 - Microprocessor with 5 address bits ($A_4 \dots A_0$) $\rightarrow 2^5 = 32$ bytes addressable
 - Memory chip: 4 x 8 (4 bytes) \rightarrow Decodes two address bits ($A_1 A_0$)
 - Can address up to 8 chips (decode address bits ($A_4 A_3 A_2$) for chip enable)



Typical generic SRAM



CE' = **chip enable**: initiate memory access when active

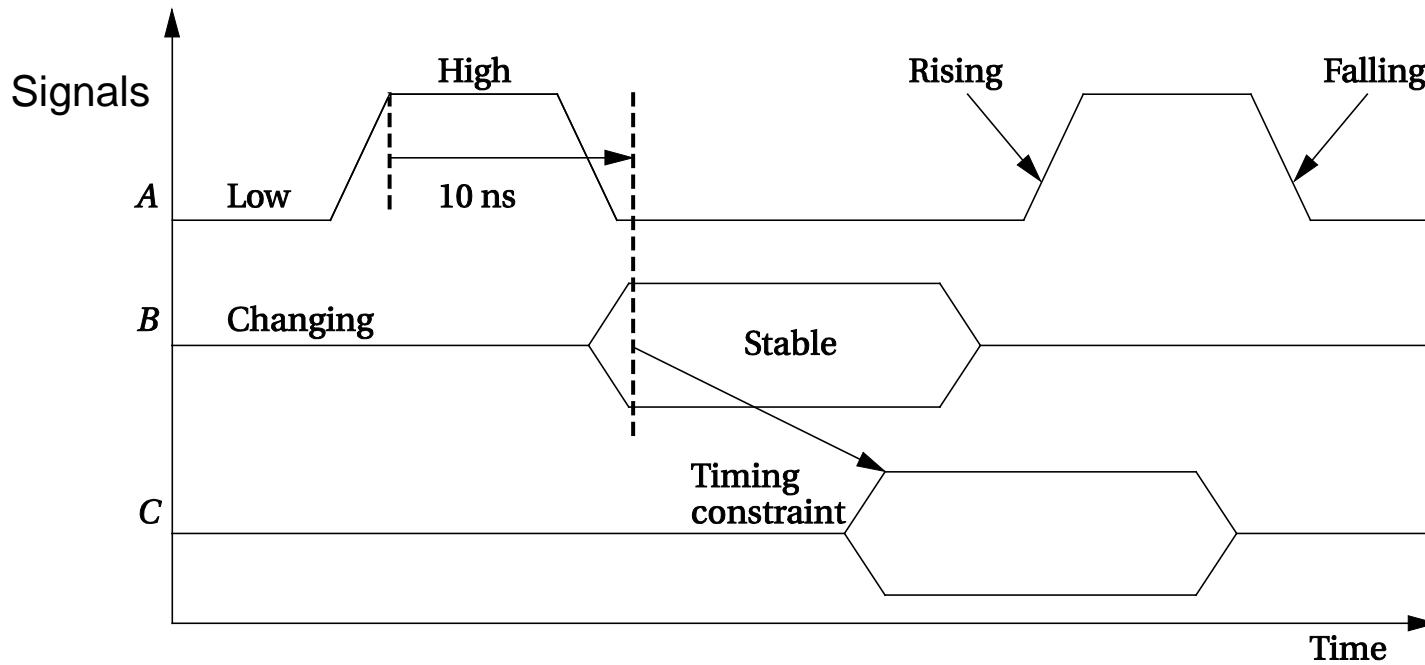
OE' = **output enable**: drive Data lines when active

WE' = **write enable**: update SRAM contents with Data

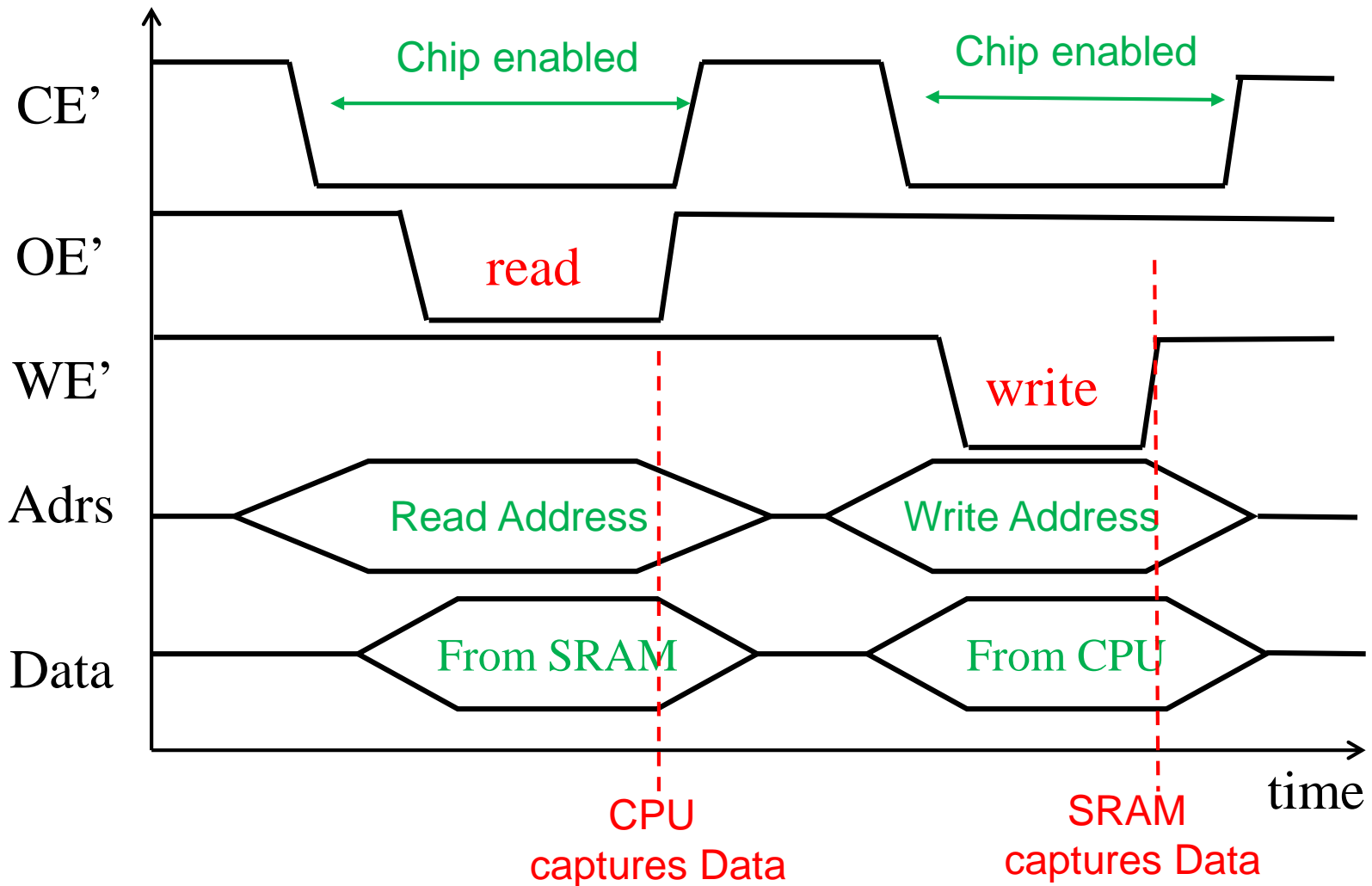
(May have one R/W' signal instead of OE' and WE')

Multi-byte data bus devices have a **byte-enable** signal for each byte.

Timing diagrams

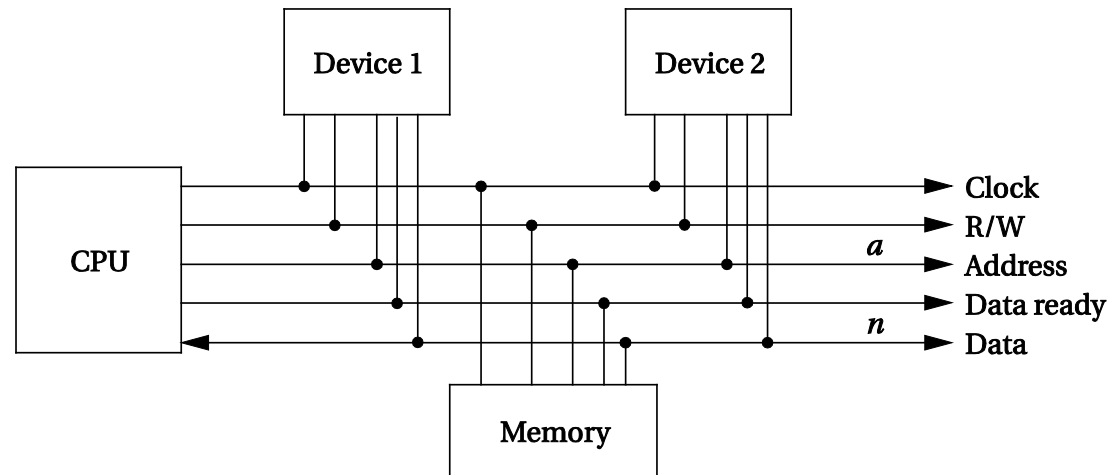


Generic SRAM timing



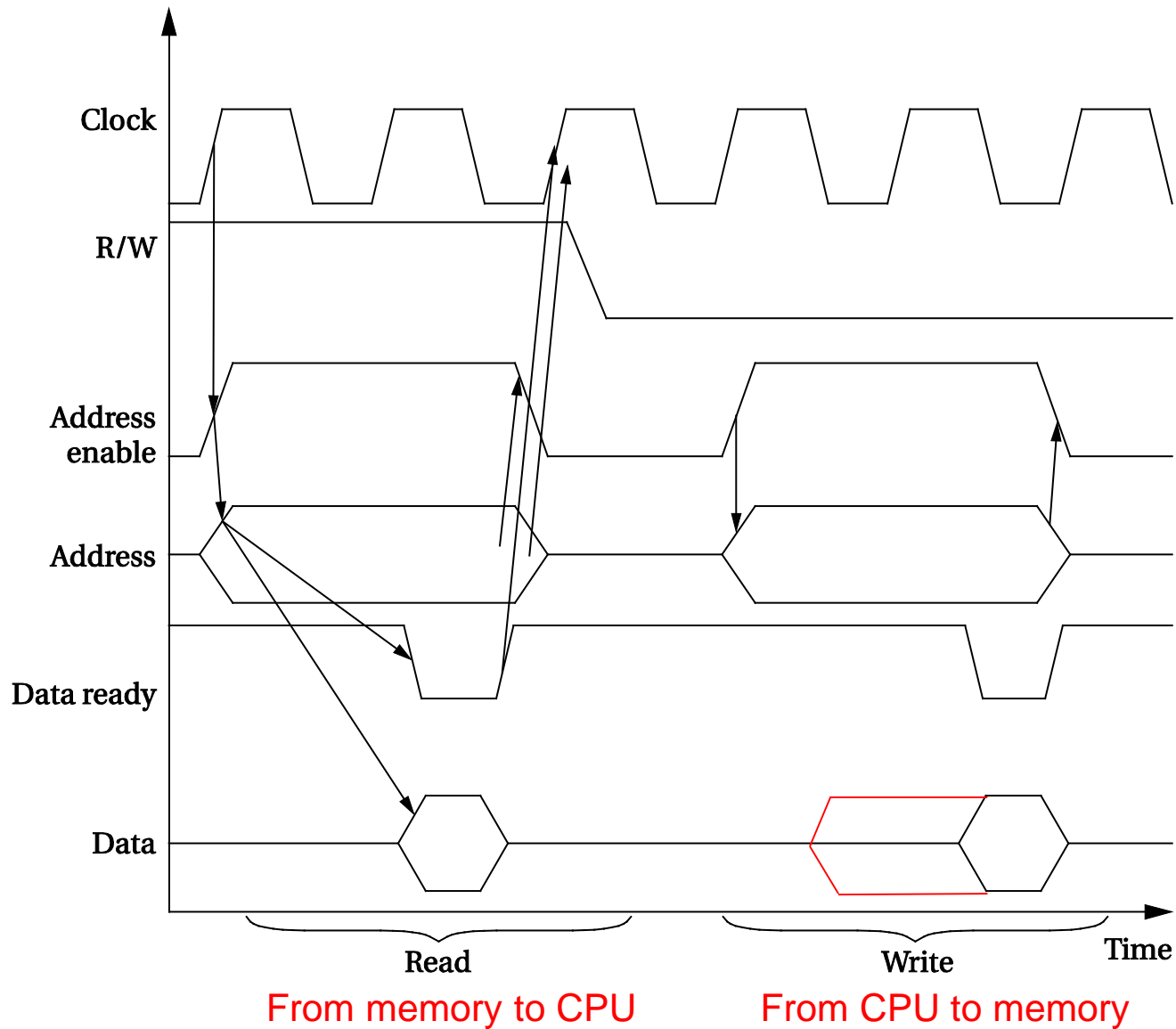
Microprocessor buses

- Mechanism for communication with memories and I/O devices
 - signal wires with designated functions
 - protocol for data transfers
 - electrical parameters (voltage, current, capacitance, etc.)
 - physical design (connectors, cables, etc.)
- Clock for synchronization.
- R/W true when reading ** (R/W' false when reading).
- Address = bundle of a address lines.
- Data = bundle of n data lines.
- “Data ready” => addressed device ready to complete the read/write

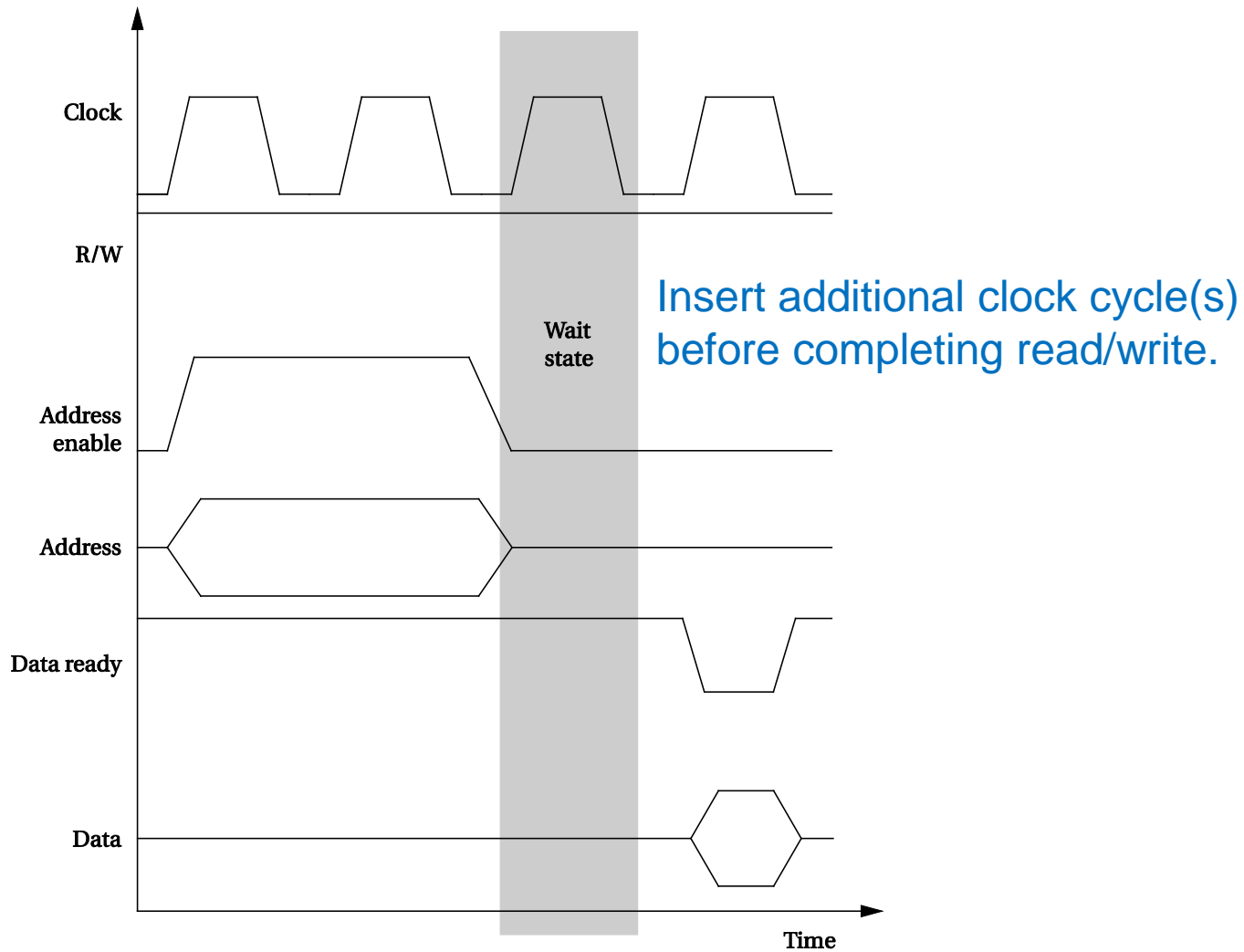


** Instead of R/W' some CPUs have separate RD' and WR'

Typical bus read and write timing



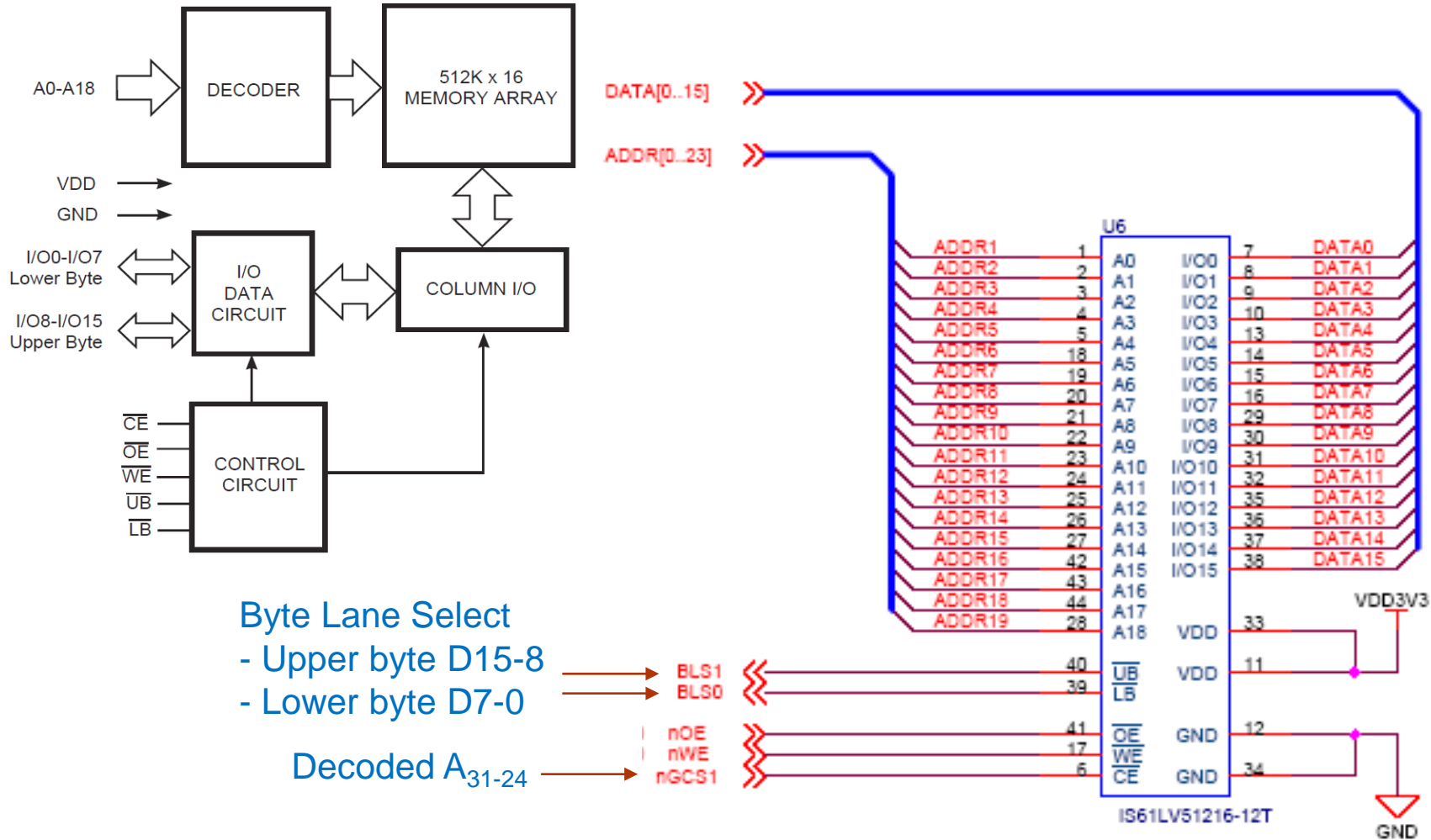
Bus wait state



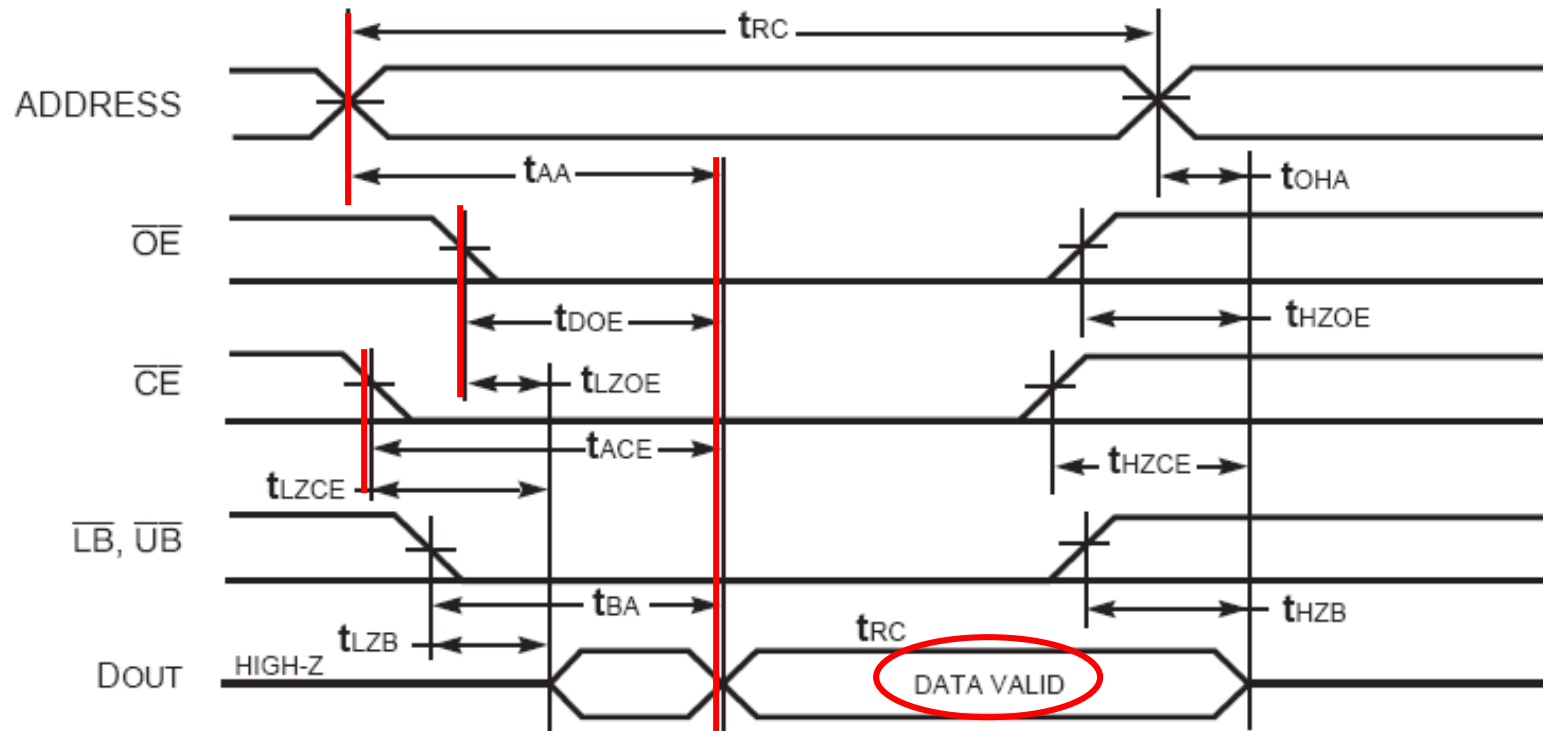
IS61LV51216-12T: 512K x 16 SRAM

2^{19}

1024K Bytes = 1M Byte



ISSI IS61LV51216 SRAM read cycle



Timing Parameters:
Max data valid times
following activation of
Address, CE, OE

Symbol	Parameter	-8		-10		-12		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
t _{RC}	Read Cycle Time	8	—	10	—	12	—	ns
t _{AA}	Address Access Time	—	8	—	10	—	12	ns
t _{OHA}	Output Hold Time	3	—	3	—	3	—	ns
t _{ACE}	$\overline{\text{CE}}$ Access Time	—	8	—	10	—	12	ns
t _{DOE}	$\overline{\text{OE}}$ Access Time	—	3.5	—	4	—	5	ns

Flash memory devices

- Available in NAND or NOR structures
 - NOR flash system interface similar to SRAM (random access)
 - NAND flash system interface typically “serial” (indirect access)
- Read operations are the default, and similar to other memory devices
- Writing/erasing is initiated by writing “*commands*” to the Flash memory controller
 - Flash is programmed at system voltages.
 - Erasure time is long, and must be erased in blocks.

	SLC NAND Flash (x8)	MLC NAND Flash (x8)	MLC NOR Flash (x16)
Density	512 Mbits ¹ – 4 Gbits ²	1Gbit to 16Gbit	16Mbit to 1Gbit
Read Speed	24 MB/s ³	18.6 MB/s	103MB/s
Write Speed	8.0 MB/s	2.4 MB/s	0.47 MB/s
Erase Time	2.0 mSec	2.0mSec	900mSec
Interface	I/O – indirect access	I/O – indirect access	Random access
Application	Program/Data mass storage	Program/Data mass storage	eXecuteInPlace

Ex: SST39VF1601- 1M x 16 NOR Flash

SST39VF3201=2M x 16 (4Mbyte: 2^{22}) / SST39VF6401=4M x 16 (8Mbyte: 2^{23})

D[15..0] »

A[20..1]** »

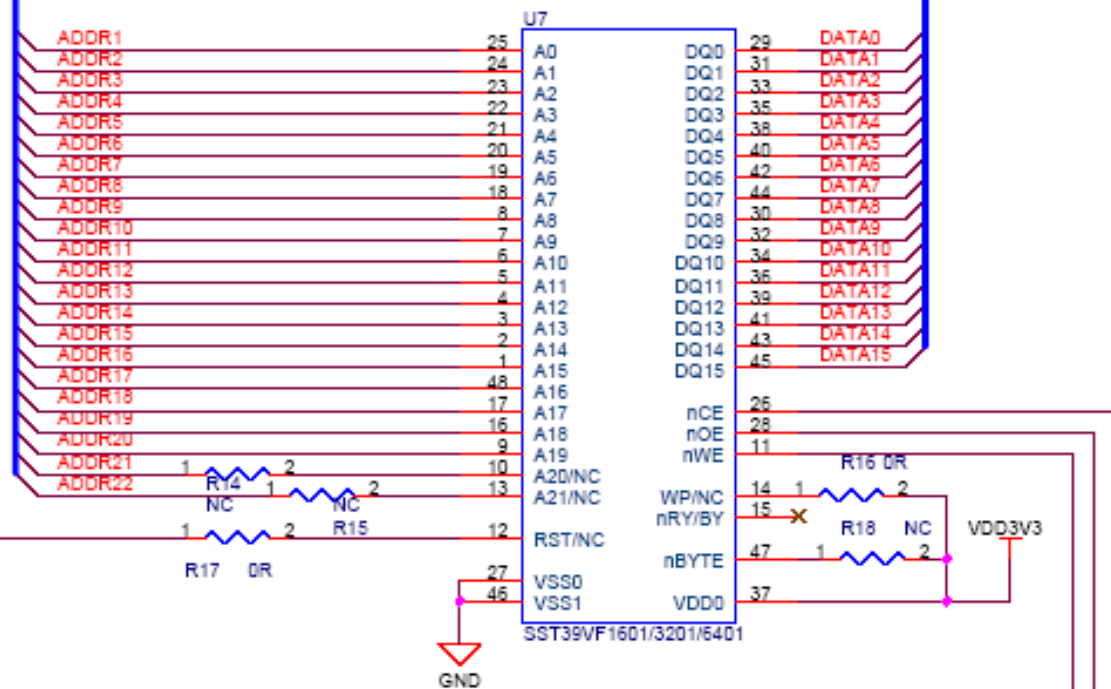
2Mbyte (2^{21}) used of
64Mbyte (2^{26}) addr. space

Therefore, 5 address bits
not decoded: A[25..21]**

A[0] is part of NBL[1:0]*

NRESET »

NOE »
NWE »
NEx »



- Byte lane selects NBL[1:0] not used: all operations are “words”
- SST39VF3201 uses A[21..1], SST39VF6401 uses A[22..1]

SST39VF1601 characteristics

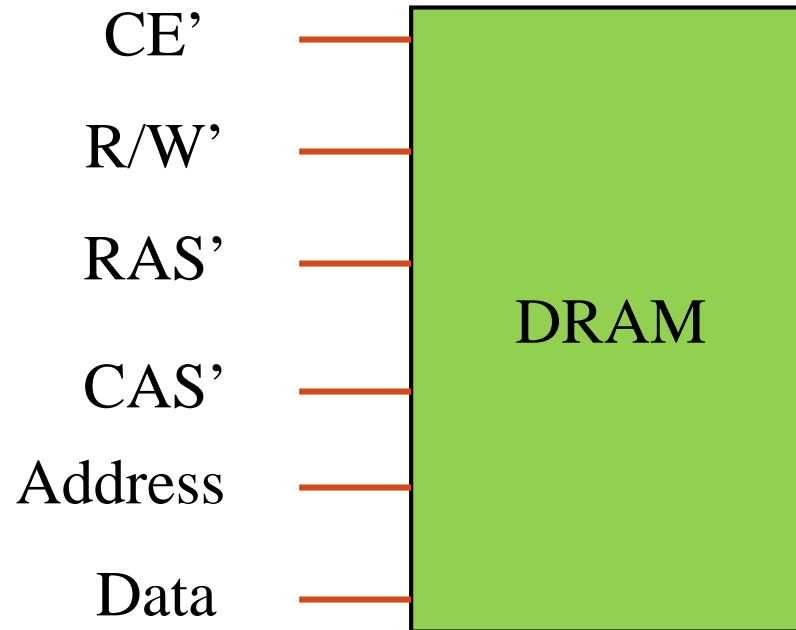
- Organized as 1M x 16
 - 2K word sectors, 32K word blocks
- Performance:
 - Read access time = 70ns or 90ns
 - Word program time = 7us
 - Sector/block erase time = 18ms
 - Chip erase time = 40ms
- Check status of write/erase operation via read
 - DQ7 = complement of written value until write complete
 - DQ7=0 during erase, DQ7=1 when erase done

SST39VF1601 command sequences

(assert WE# and CE# to write commands)

Command Sequence	1st Bus Write Cycle		2nd Bus Write Cycle		3rd Bus Write Cycle		4th Bus Write Cycle		5th Bus Write Cycle		6th Bus Write Cycle	
	Addr ¹	Data ²	Addr ¹	Data ²	Addr ¹	Data ²	Addr ¹	Data ²	Addr ¹	Data ²	Addr ¹	Data ²
Word-Program	5555H	AAH	2AAAH	55H	5555H	A0H	WA ³	Data				
Sector-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	SA _X ⁴	30H
Block-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	BA _X ⁴	50H
Chip-Erase	5555H	AAH	2AAAH	55H	5555H	80H	5555H	AAH	2AAAH	55H	5555H	10H
Erase-Suspend	XXXXH	B0H										
Erase-Resume	XXXXH	30H										
Query Sec ID ⁵	5555H	AAH	2AAAH	55H	5555H	88H						
User Security ID Word-Program	5555H	AAH	2AAAH	55H	5555H	A5H	WA ⁶	Data				
User Security ID Program Lock-Out	5555H	AAH	2AAAH	55H	5555H	85H	XXH ⁶	0000H				
Software ID Entry ^{7,8}	5555H	AAH	2AAAH	55H	5555H	90H						
CFI Query Entry	5555H	AAH	2AAAH	55H	5555H	98H						
Software ID Exit ^{9,10} /CFI Exit/Sec ID Exit	5555H	AAH	2AAAH	55H	5555H	F0H						
Software ID Exit ^{9,10} /CFI Exit/Sec ID Exit	XXH	F0H										

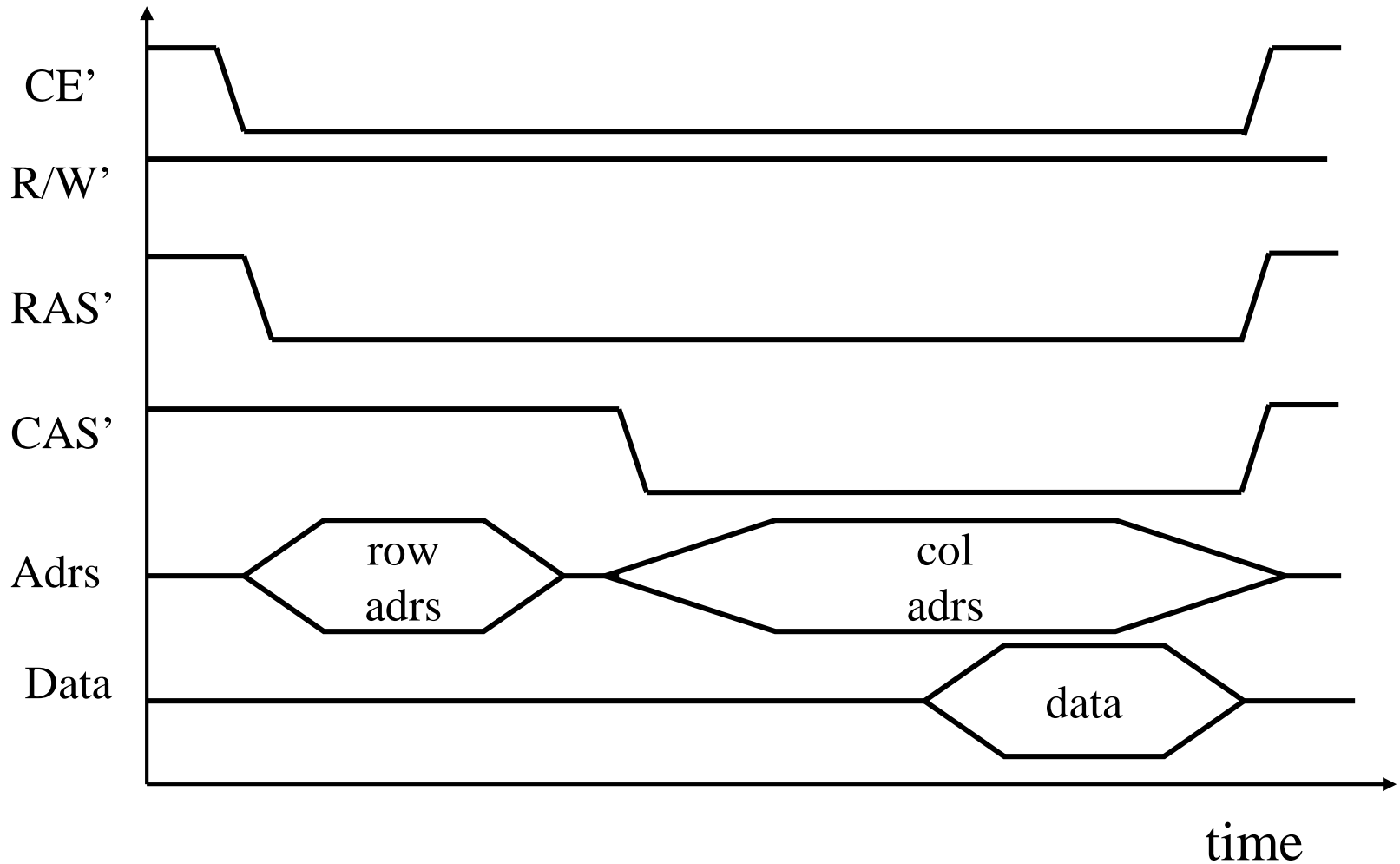
Generic DRAM device



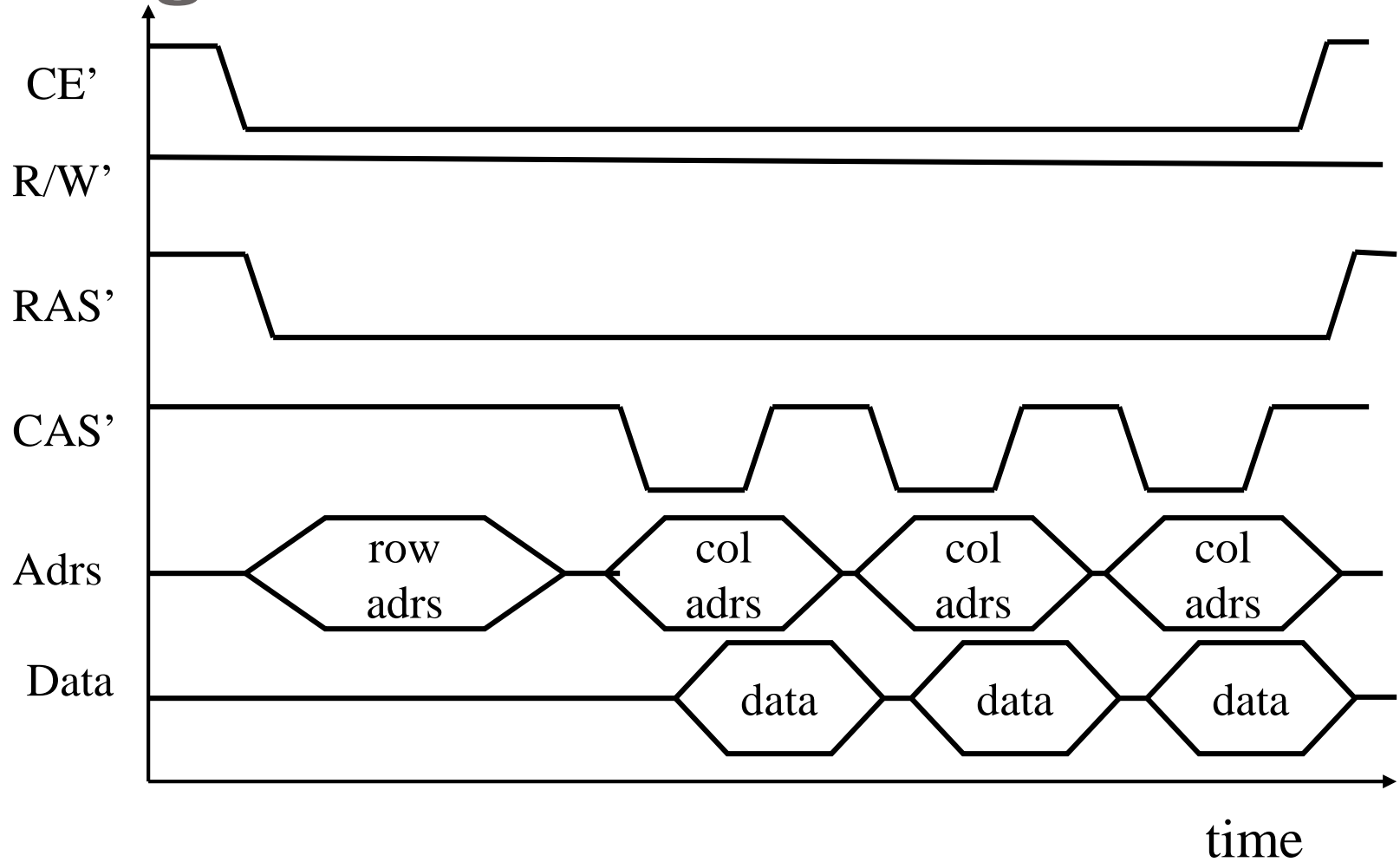
RAS' = **Row Address Strobe**: row# on Address inputs

CAS' = **Column Address Strobe**: column# on Address inputs

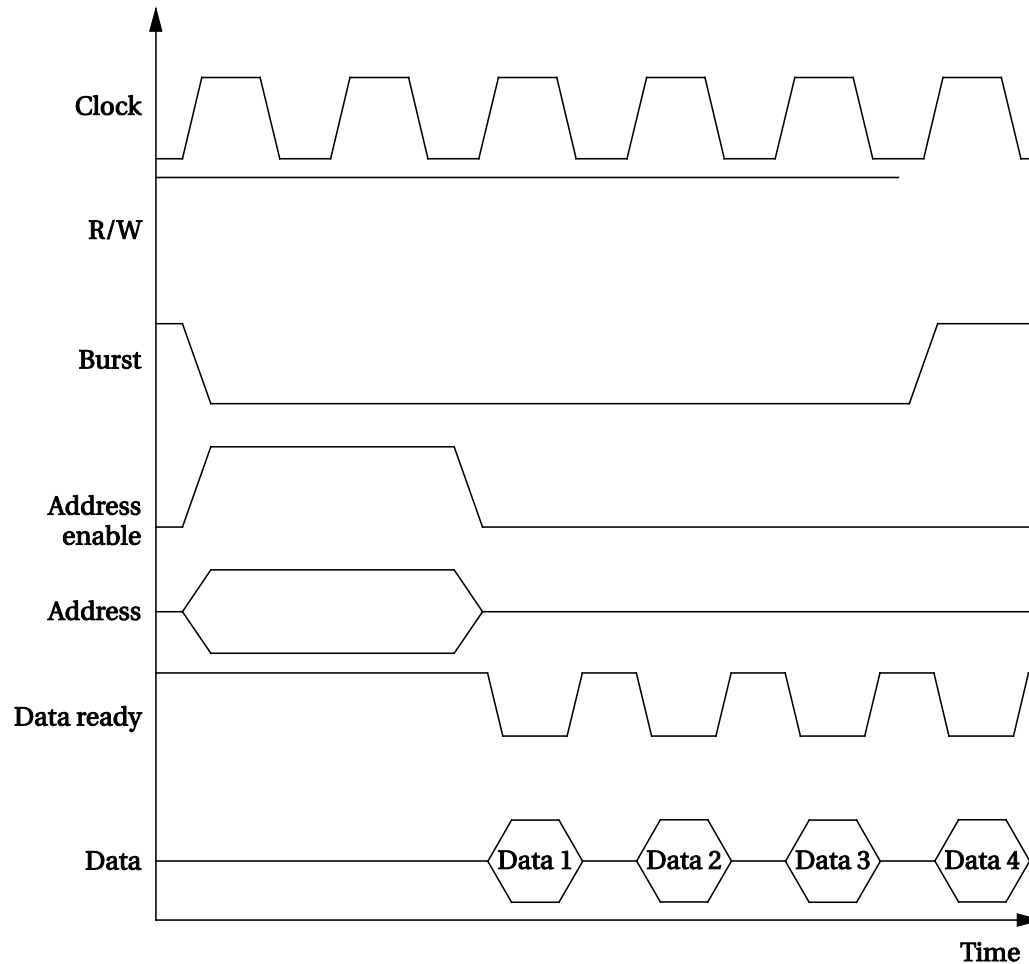
Generic DRAM timing



Page mode access



Bus burst read (if supported)



Dynamic RAM refresh

- Value decays in approx. 1-4 ms.
- Refresh value by reading it
 - Read row of bits and then copy back
 - Can't access memory during refresh.
- RAS-only refresh
- CAS-before-RAS refresh.
- Hidden refresh.

4 Mbyte DRAM: Refresh every 4 msec

Organized as **2048 rows** x 2048 columns → 2048 refreshes

Assume 1 row refresh → 80 nsec

$$\frac{2048 \times 80 \times 10^{-9}}{4 \times 10^{-3}} \cong 0.041 \rightarrow 4.1\% \text{ of time spent refreshing}$$

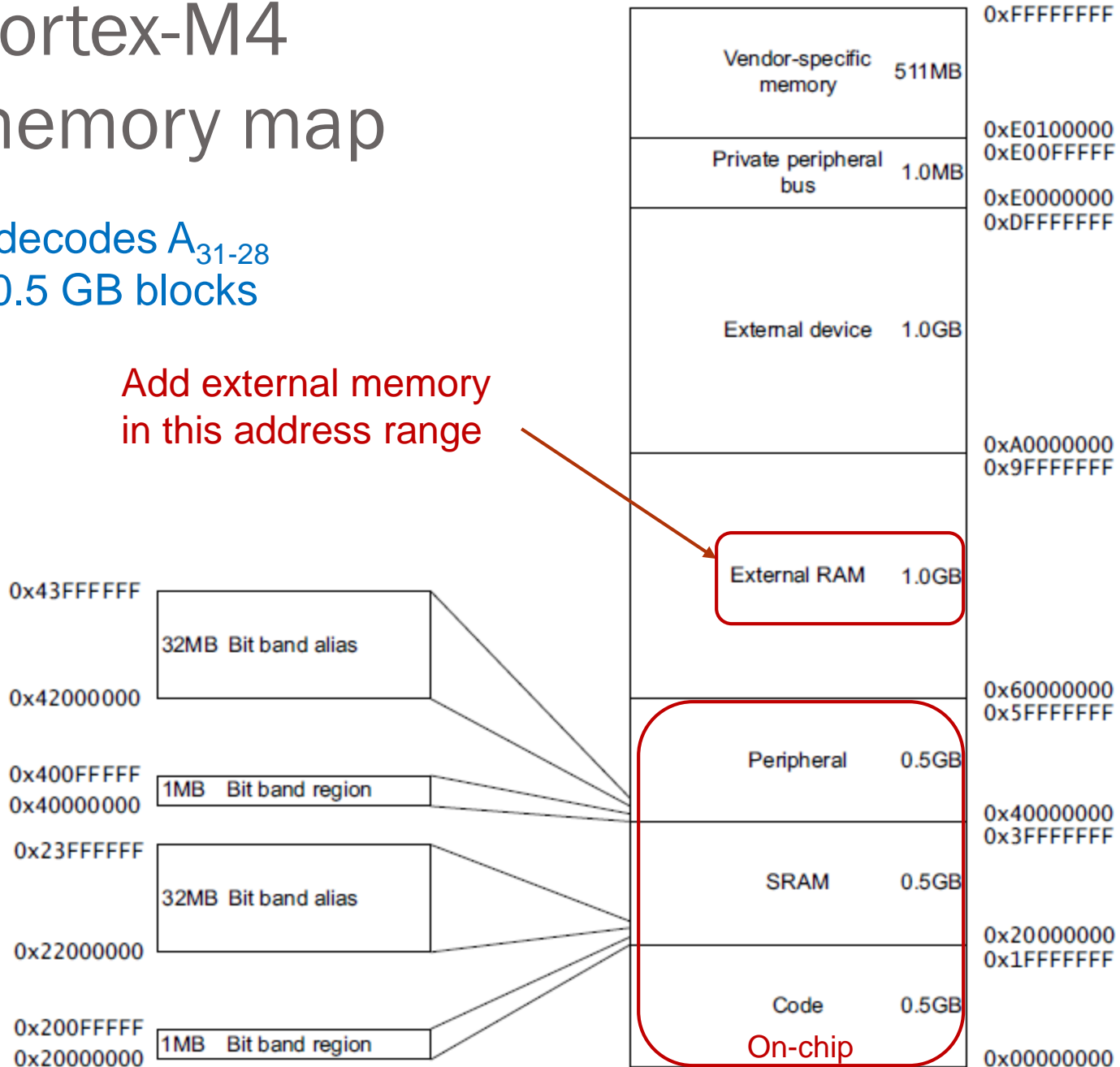
Other DRAM forms

- Extended data out (EDO): improved page mode access.
- Synchronous DRAM: clocked access for pipelining.
- Double Data Rate (DDR) – transfer on both edges of clock
 - DDR-1, DDR-2, DDR-3 support increasingly higher bandwidths
- Rambus: highly pipelined DRAM.

Cortex-M4 memory map

uC decodes A_{31-28}
for 0.5 GB blocks

Add external memory
in this address range

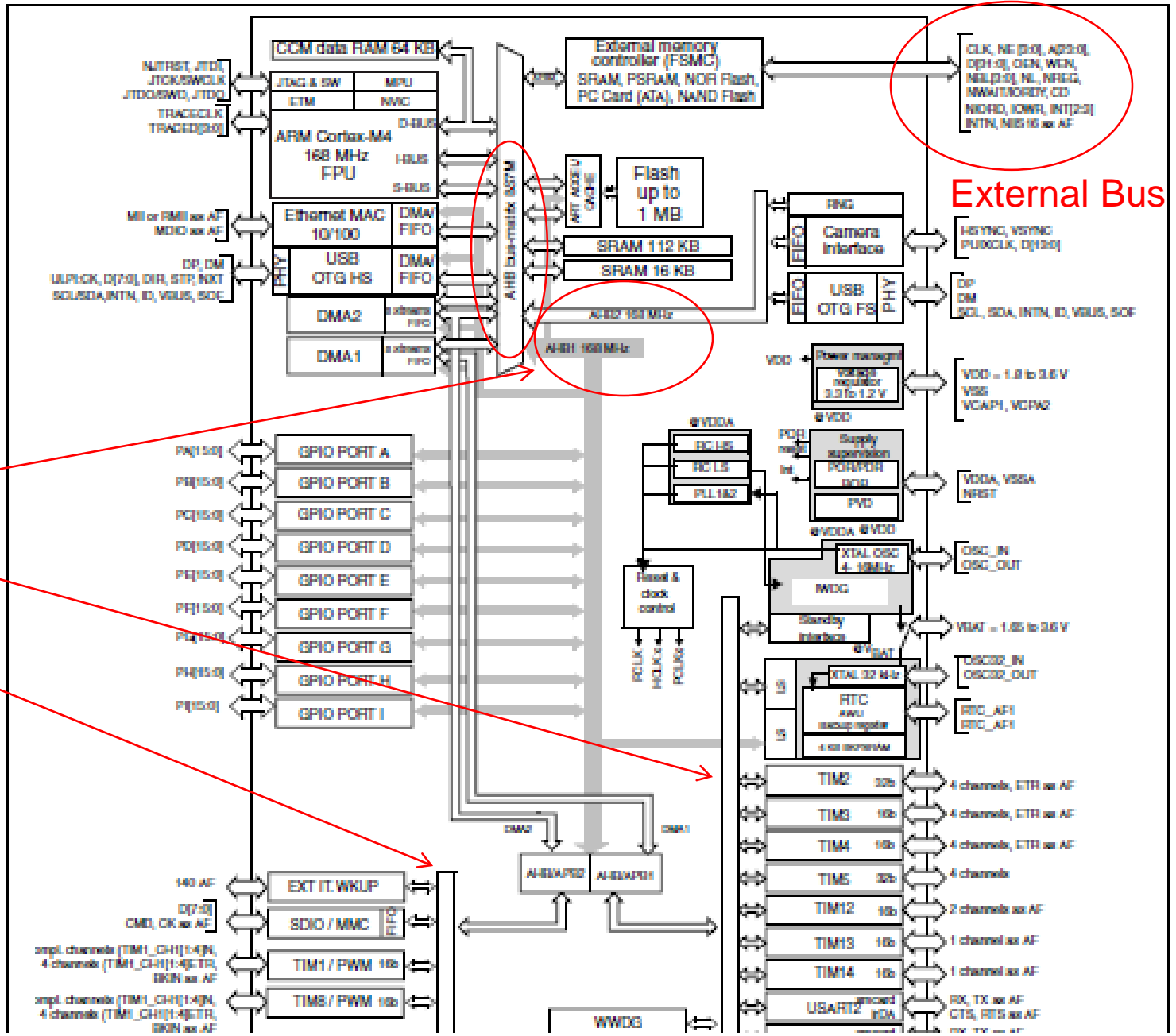


STM32F407 Microcontroller

AHB 168MHz

APB 142MHz

APB 84MHz

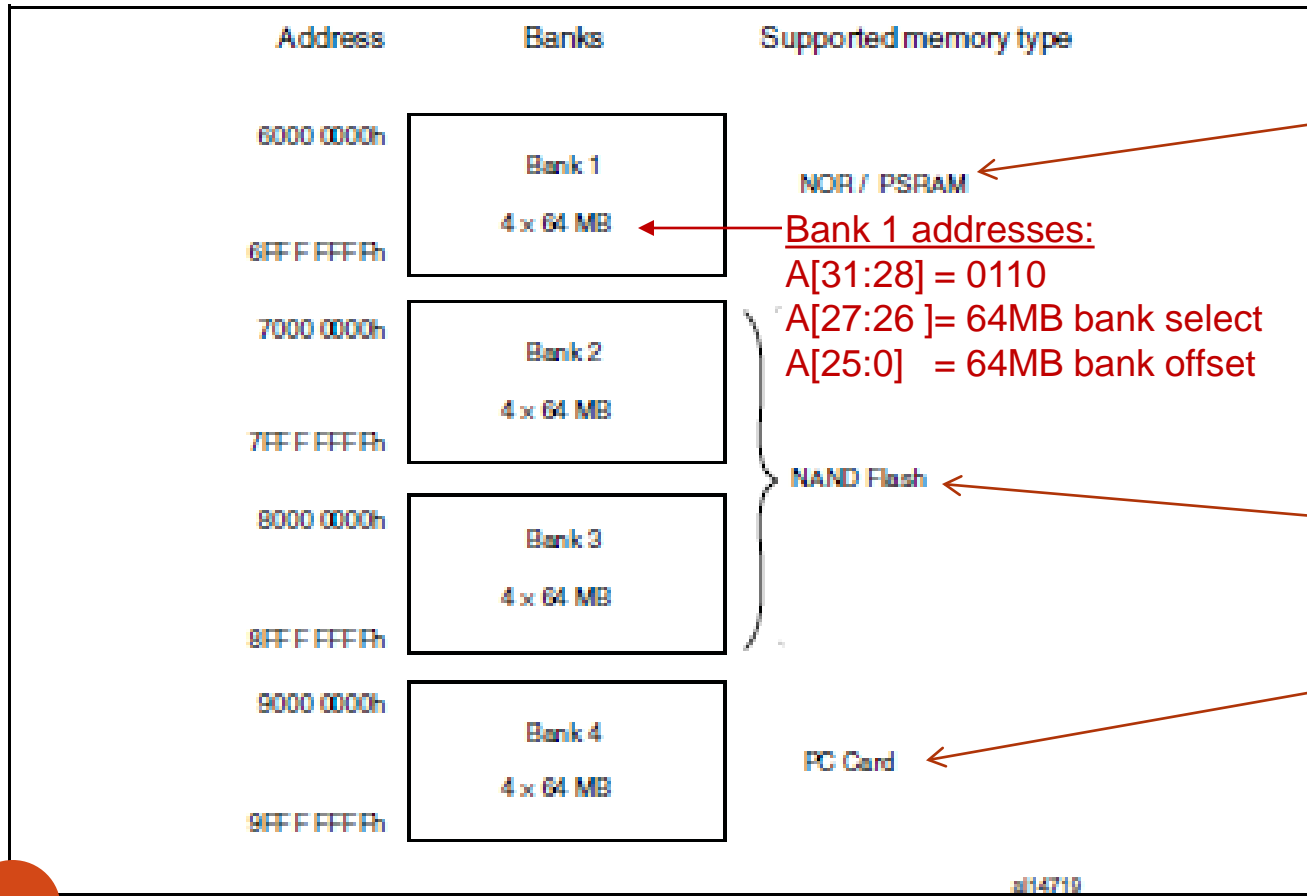


External Bus

STM32 Flexible Static Memory Controller (FSMC)

- STM32F4xx Tech. Ref. Manual, Chap. 36

- Control external memory on AHB bus in 4 - 256K banks
 - Upper address bits decoded by the FSMC



Bank 1 addresses:
 $A[31:28] = 0110$
 $A[27:26] = 64\text{MB bank select}$
 $A[25:0] = 64\text{MB bank offset}$

Static memory-mapped devices:

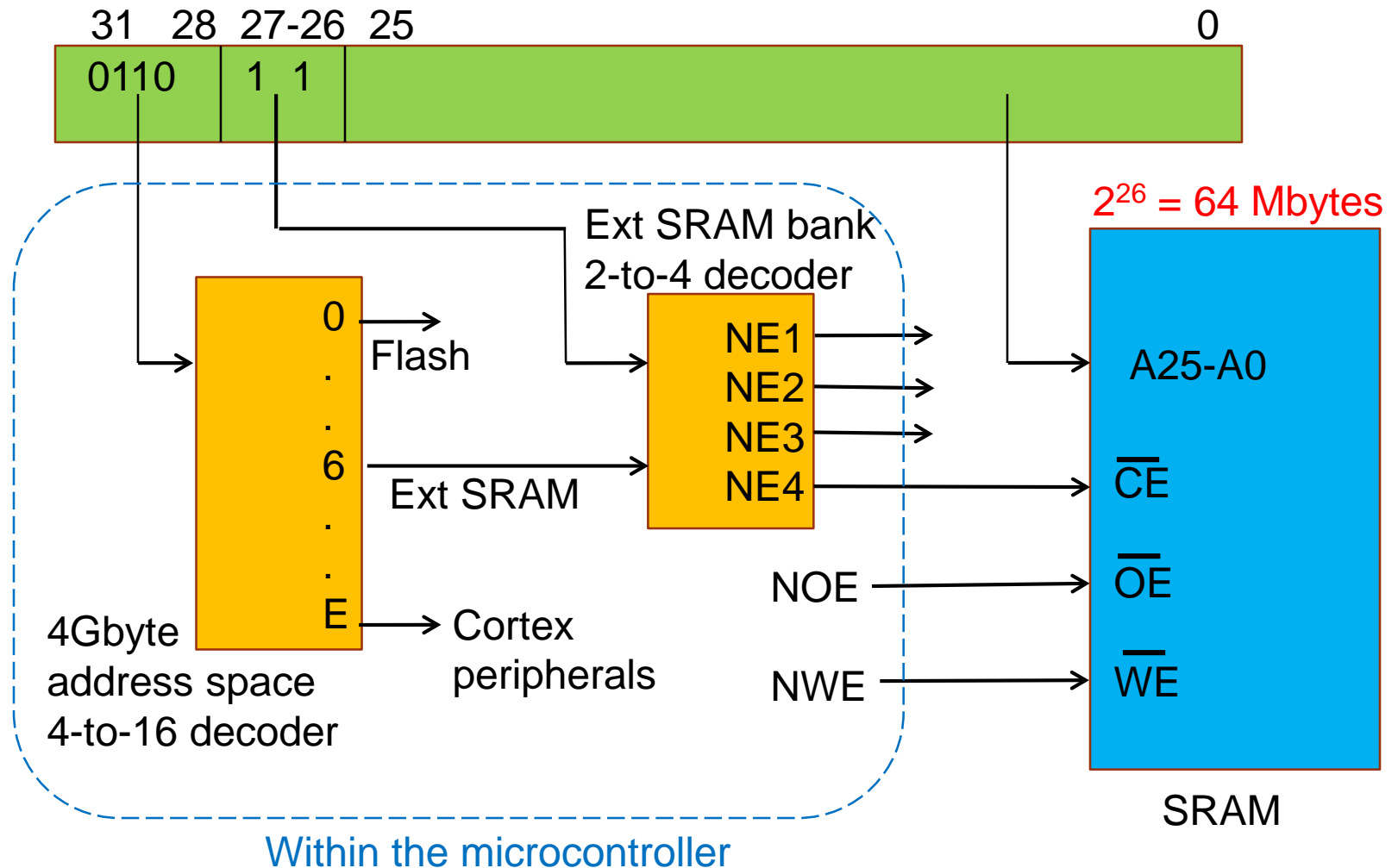
- * SRAM
- * Pseudo-Static RAM
- * NOR flash

2 banks NAND flash

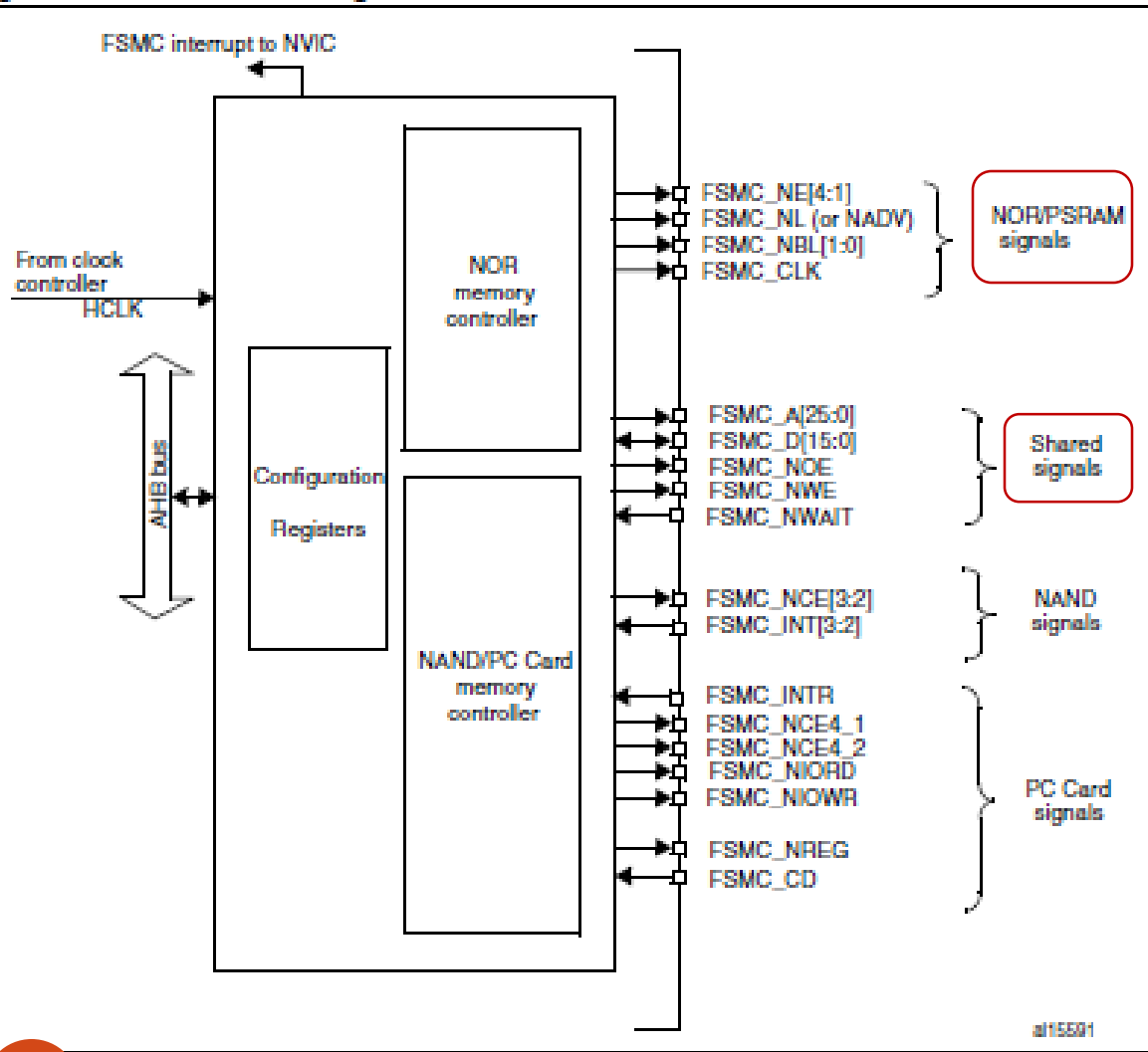
16-bit PC-Card devices

Example SRAM address decoding

SRAM/NE4 Addresses: [0x6C00 0000 ... 0x6F00 0000]



FSMC block diagram



“N” = “negative” (active low)

NE[4:1] = NOR/PSRAM enable

- NE[1]: A[27:26]=00
- NE[2]: A[27:26]=01
- NE[3]: A[27:26]=10
- NE[4]: A[27:26]=11

NL = address latch/advance

NBL = byte lane

CLK for sync. Burst

A[25:0] = Address bus

D[15:0] = Data bus**

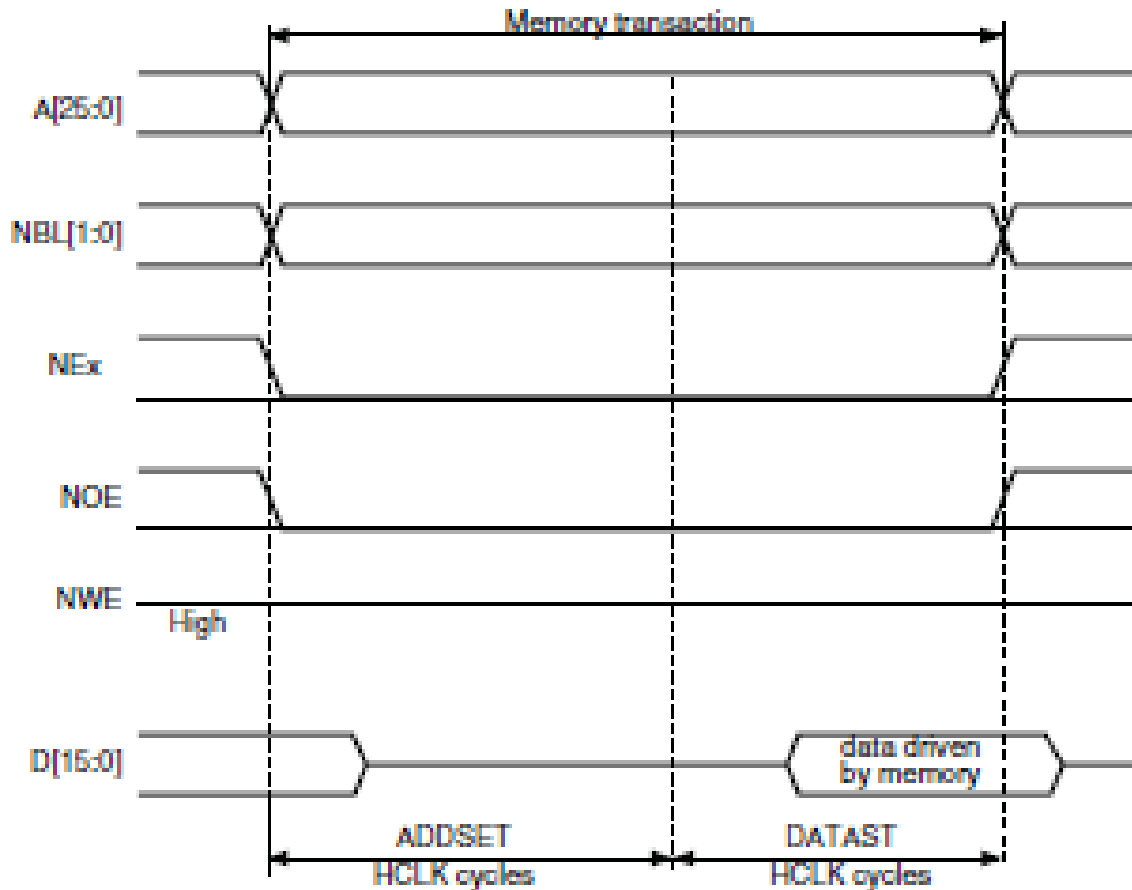
NOE = output enable

NWE = write enable

NWAIT = wait request

** Data bus = 8 or 16 bits

FSMC “Mode 1” memory read

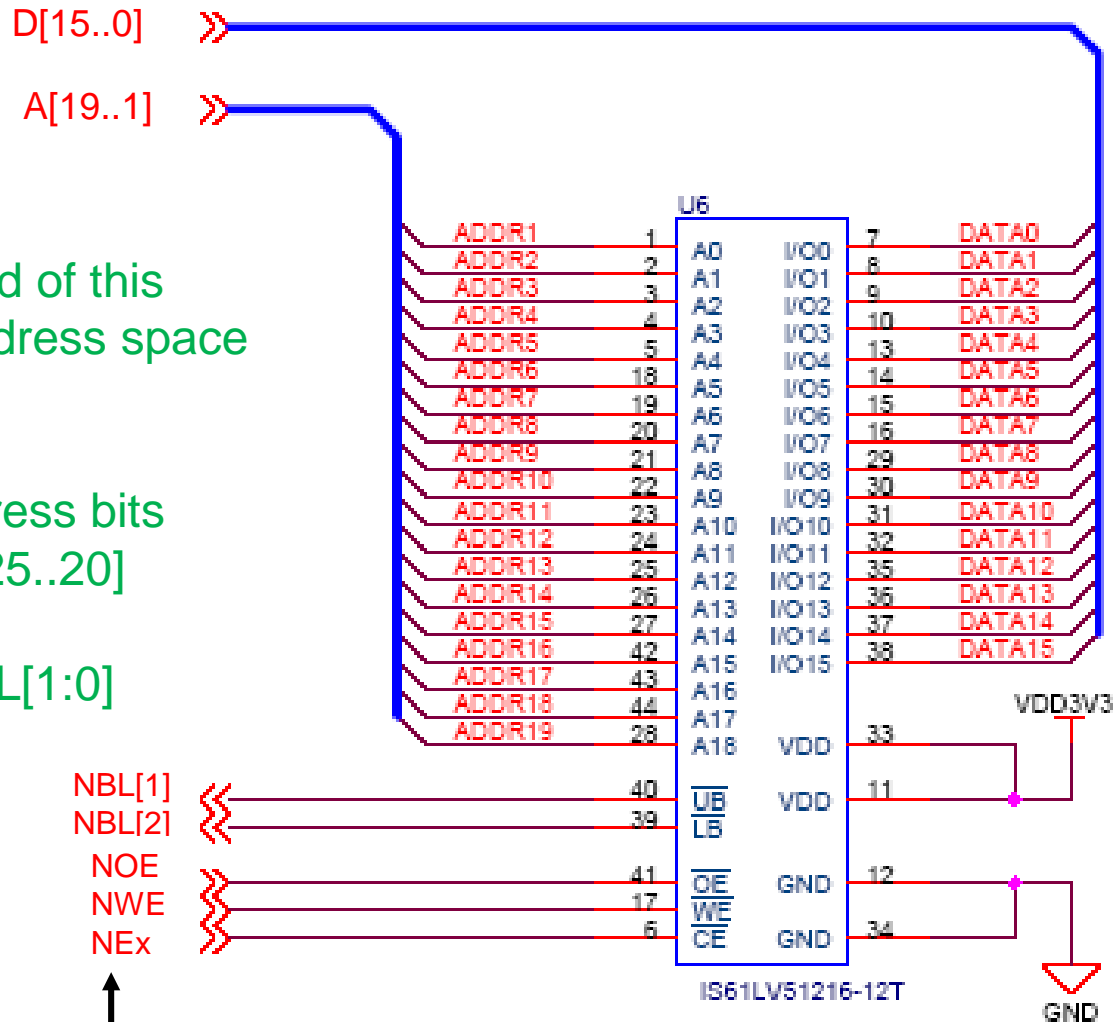


Other modes:

- * Provide ADV (address latch/advance)
- * Activate OE and WE only in DATAST
- * Multiplex A/D bits 15-0
- * Allow WAIT to extend DATAST

ADDSET/DATAST programmed in chip-select timing register (HCLK = AHB clock)

Example: 512K x 16 SRAM (1 Mbyte)



1Mbyte (2^{20}) used of this
64Mbyte (2^{26}) address space
for NEx

Therefore, 6 address bits
not decoded: A[25..20]

A[0] is part of NBL[1:0]

NBL[1]
NBL[2]
NOE
NWE
NEx

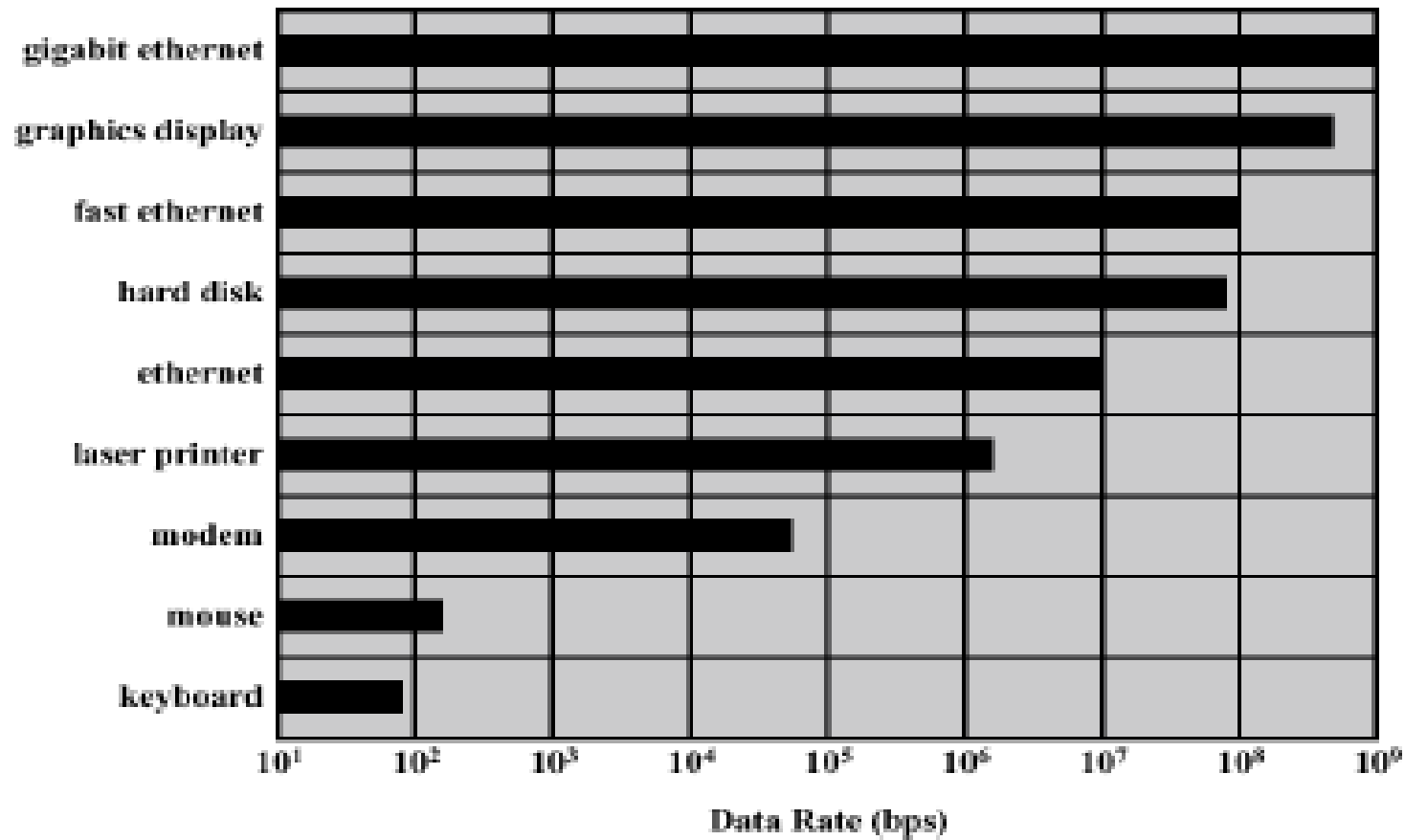


Microcontroller decodes upper address bits – ADDR[31..26] – for NEx

CPU Bus Types

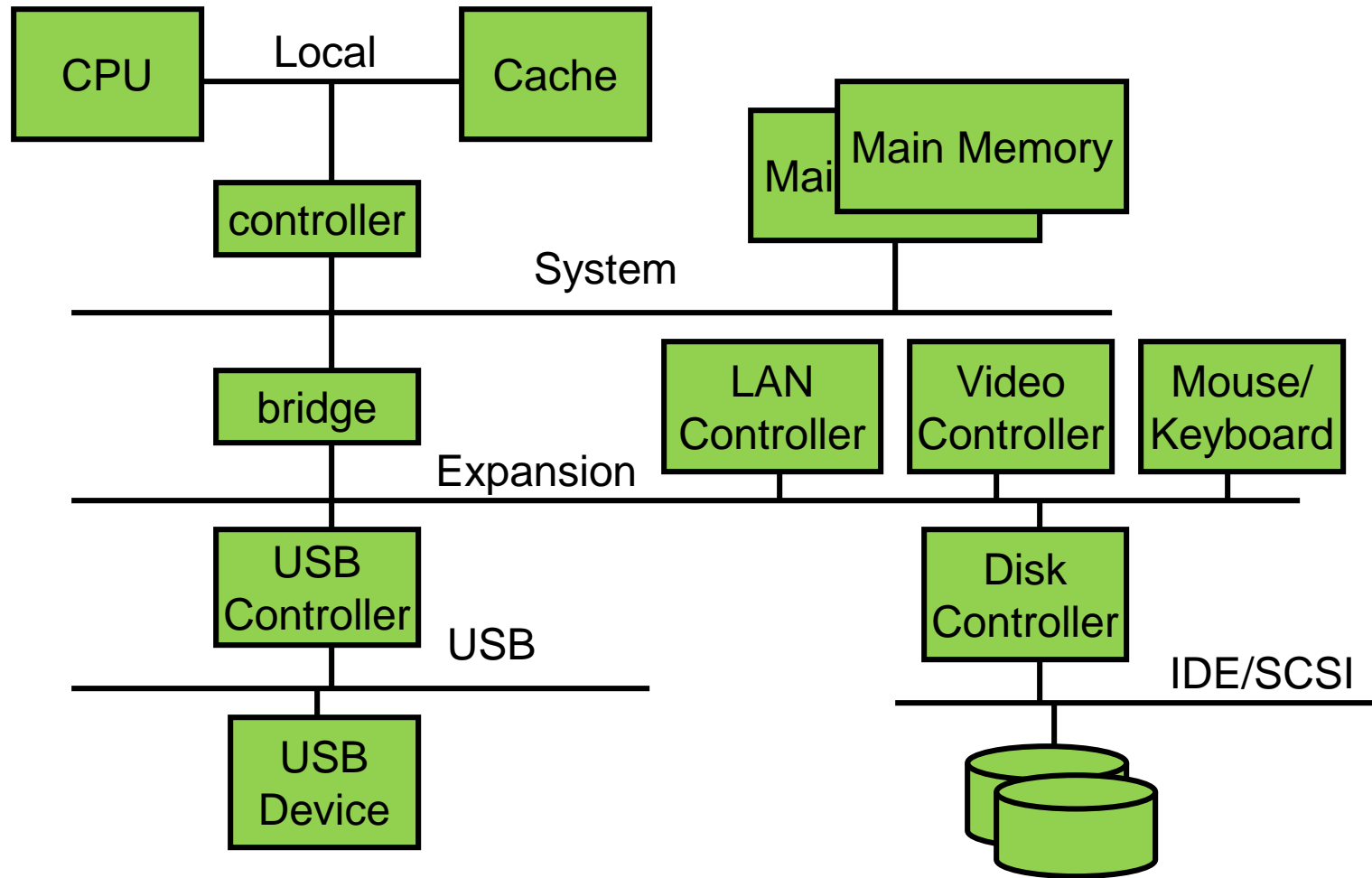
- Synchronous vs. Asynchronous
 - Sync: all op's synchronized to a clock
 - Async: devices signal each other to indicate start/stop of operations
 - May combine sync/async (80x86 "Ready" signal)
- Data transfer types:
 - Processor to/from memory
 - Processor to/from I/O device
 - I/O device to/from memory (DMA)
- Data bus types
 - Parallel (data bits transferred in parallel)
 - Serial (data bits transferred serially)

Typical bus data rates

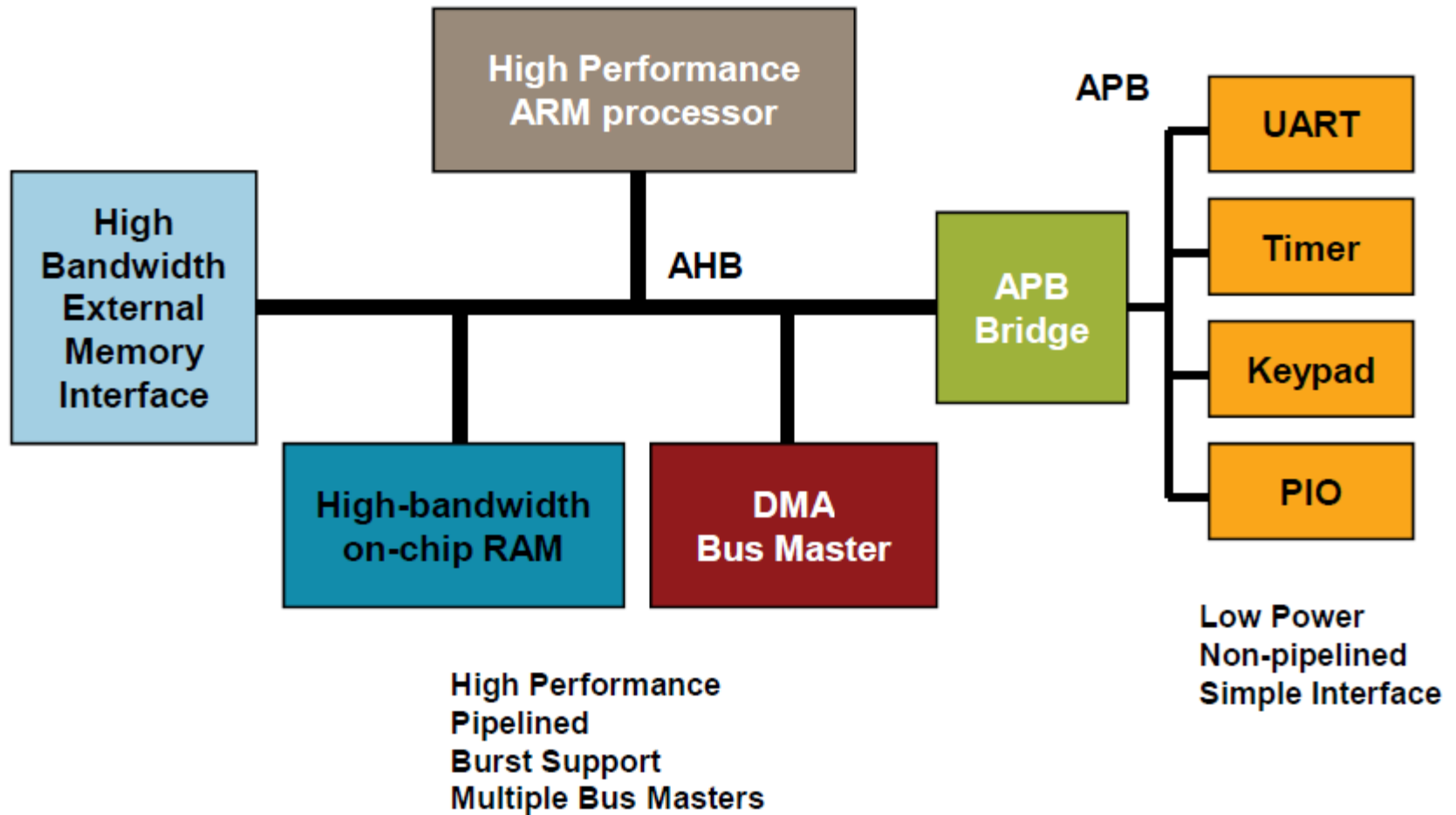


Source: Peter Cheung "Computer Architecture & Systems Course Notes"

Hierarchical Bus Architecture

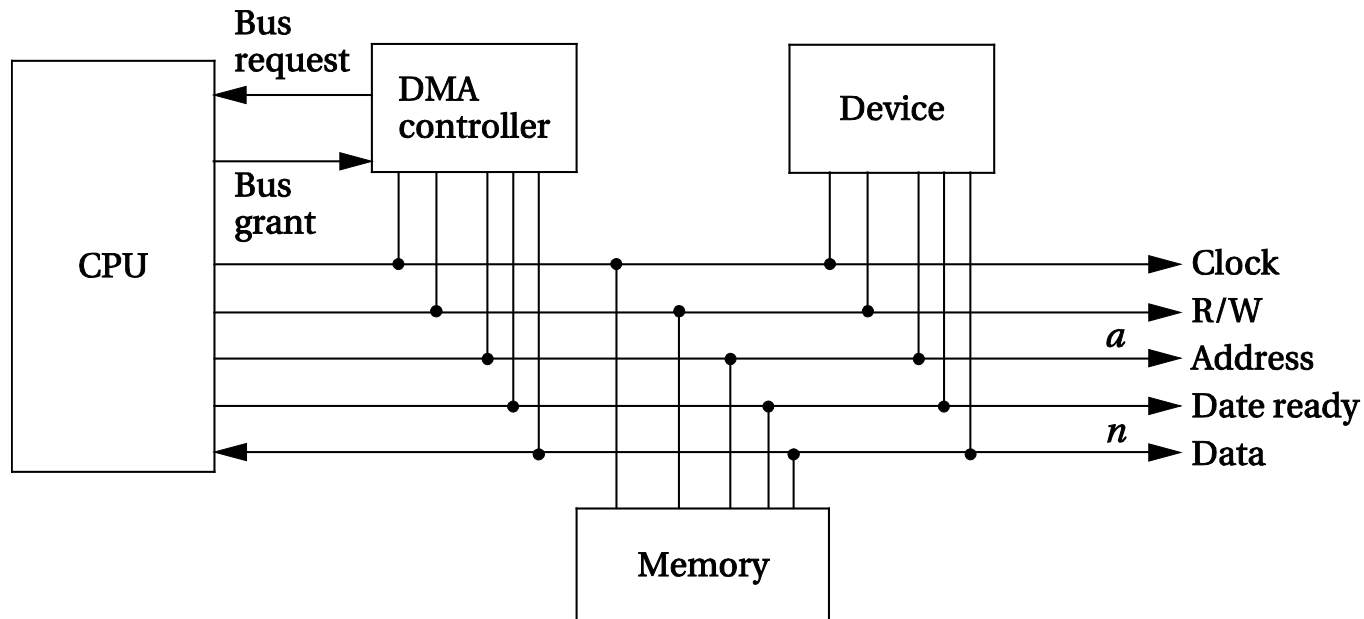


Example ARM System



DMA

- ▶ Direct memory access (DMA) performs data transfers without executing instructions.
 - ▶ CPU sets up transfer by programming the DMA controller.
 - ▶ DMA engine fetches, writes.
- ▶ DMA controller is a separate unit – can become bus master.



Bus mastership

- Bus master controls operations on the bus.
- By default, CPU is bus master and initiates transfers.
- Other devices may request bus mastership.
 - Separate set of handshaking lines.
 - CPU can't use bus when it is not master.
- Bus mastership protocol:
 - Bus request – a device requests bus mastership from CPU
 - Bus grant – CPU relinquishes and grants mastership to device
- Situations for multiple bus masters:
 - DMA data transfers
 - Multiple CPUs with shared memory
 - One CPU might be graphics/network processor

DMA operation

- ▶ CPU configures DMA controller registers for:
 - ▶ peripheral address, memory start address, #xfers, direction (P->M or M->P)
- ▶ Peripheral issues DMA request to DMA controller.
- ▶ DMA controller takes bus mastership from CPU
- ▶ Once DMA is bus master, it transfers automatically.
 - ▶ Memory address incremented and count decremented for each transfer.
 - ▶ May run continuously until complete.
 - ▶ May use every n^{th} bus cycle.

Bus master request

