EDA322
Digital Design

Lecture 14:

Interconnect and Memories

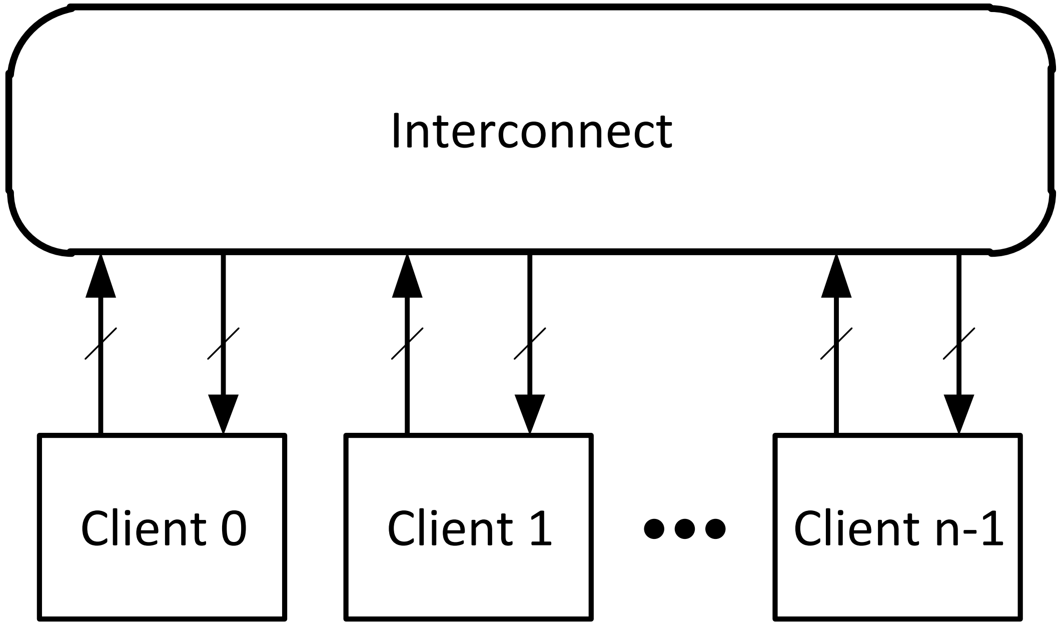
Ioannis Sourdis

Outline of Lecture 14

- Interconnect
- Memories
  - Standard Memory organization
  - RAM, ROM, etc.

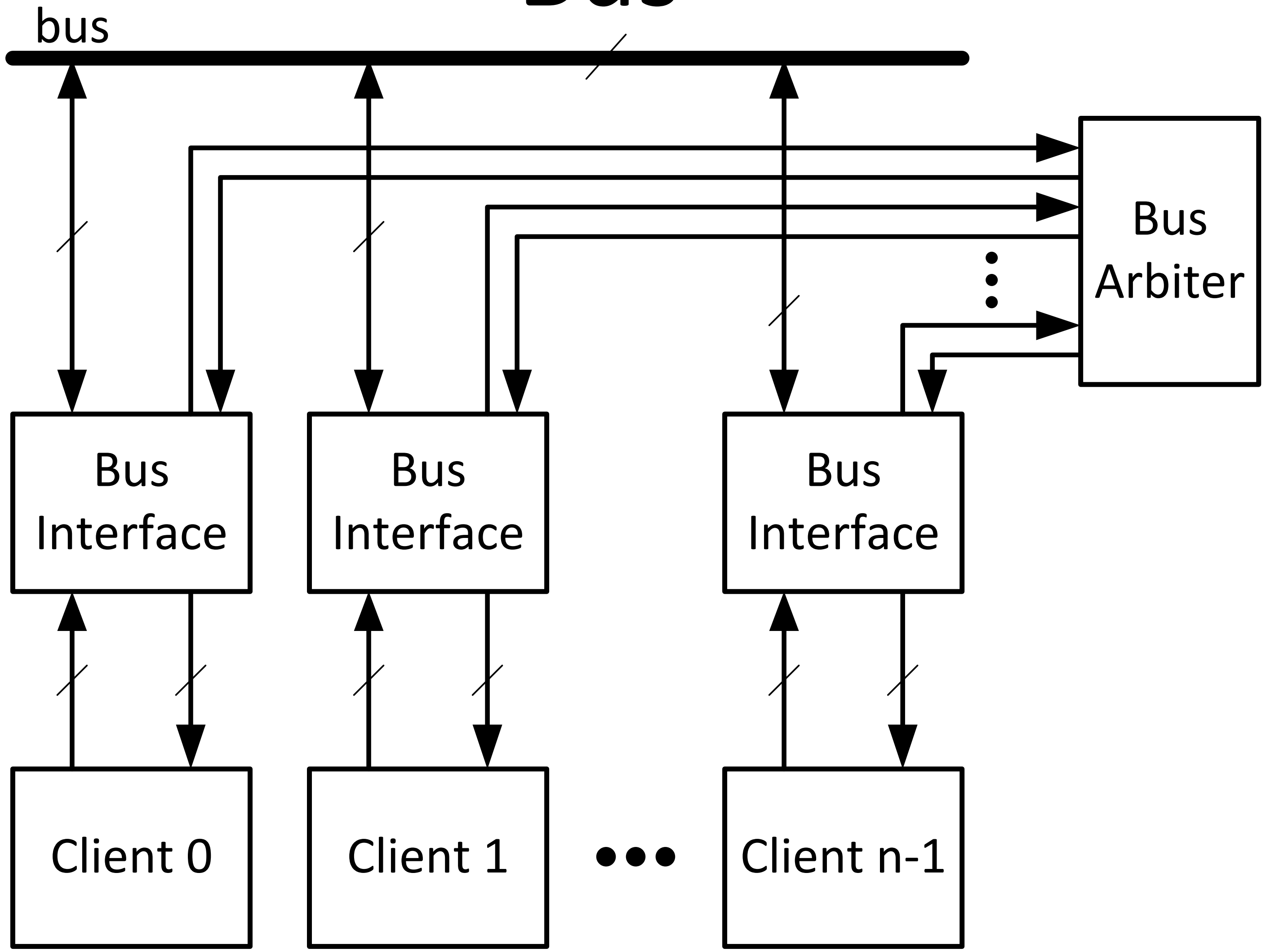
Interconnect

Interconnect

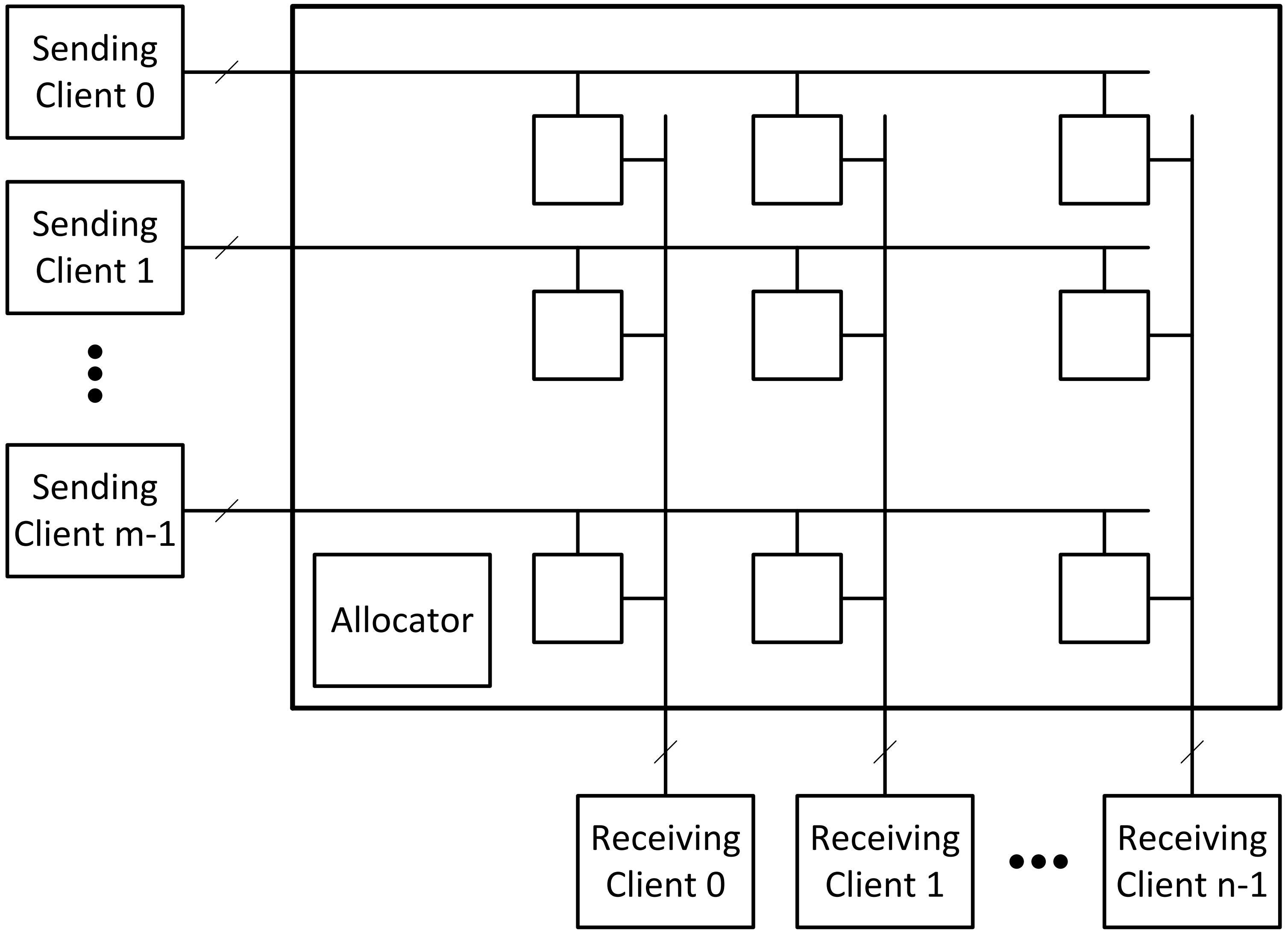


- Many clients need to communicate
- Ad-hoc point-to-point wiring or shared interconnect
- Like a telephone exchange

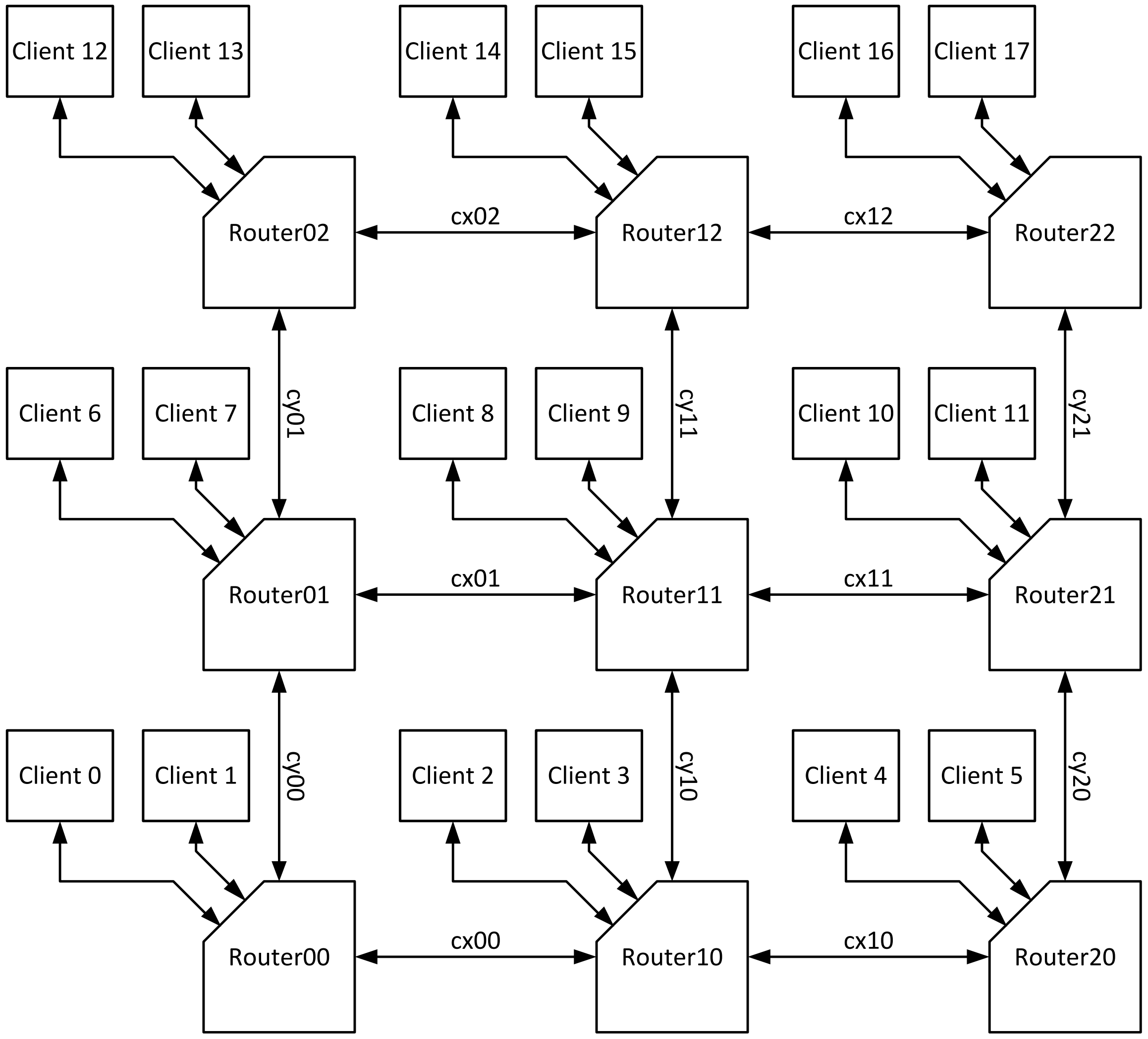
Bus


bus
Bus Arbiter
Bus Interface
Bus Interface
Bus Interface
Client 0
Client 1
Client n-1

Crossbar Switch



Interconnection Networks



Interconnection Networks

Cycle	0	1	2	3	4	5	6	7	8	9	10	11
Client 0 output	▭											
Router00		▭										
cx00e			▭									
Router10				▭	▭	▭						
cx10e							▭					
Router20								▭	▭			
cy20n										▭		
Router21											▭	
Client 11 input												▭

What factors determine which interconnect solution you pick?

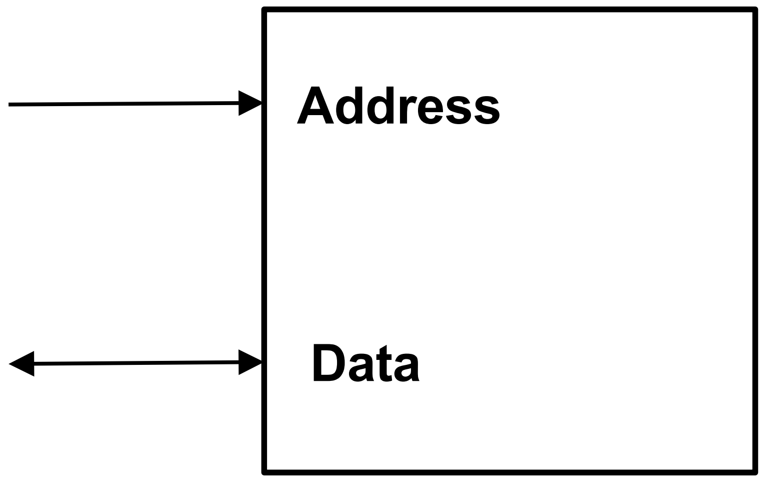
Memories

Memory Basics

- Uses:
  - data & program storage
  - general purpose registers
  - buffering
  - table lookups
  - Combinational Logic implementation
  - Whenever a large collection of state elements is required.

- Types:
  - RAM - random access memory
  - ROM - read only memory
  - EPROM, FLASH - electrically programmable read only memory

Memory



Capacity

Bandwidth

Latency

Granularity

Memory

A large number of addressable fixed size locations

- Address Space

n bits allow the addressing of 2^n memory locations.

- Example: 24 bits can address 2^{24} = 16,777,216 locations

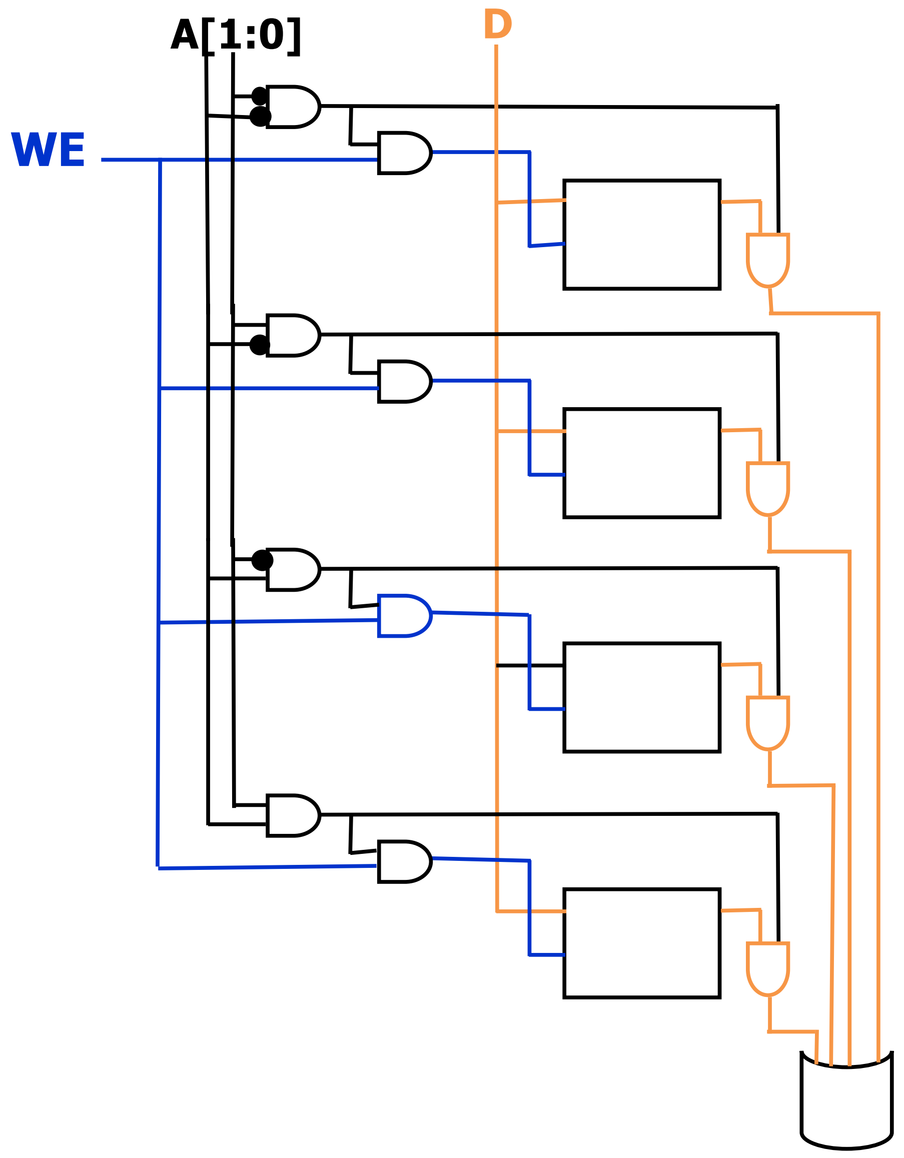
  (i.e. 16M locations).
- If each location holds 1 byte (= 8 bits) then the memory is 16MB.
- If each location holds one word (32 bits = 4 bytes) then it is 64 MB.

Memory

- Addressability
  - Computers are either byte or word addressable - i.e. each memory location holds either 8 bits (1 byte), or a full standard word for that computer (8 bits for the ChAcc processor, more typically 32 bits, though now many machines use 64 bit words).

Building a Memory

- Each bit
  - is a gated D-latch
- Each location
  - consists of w bits (here w = 1)
  - w = 8 if the memory is byte addressable
- Addressing
  - n locations means **$\log_2 n$** address bits (here 2 bits => 4 locations)
  - decoder circuit translates address into 1 of n locations



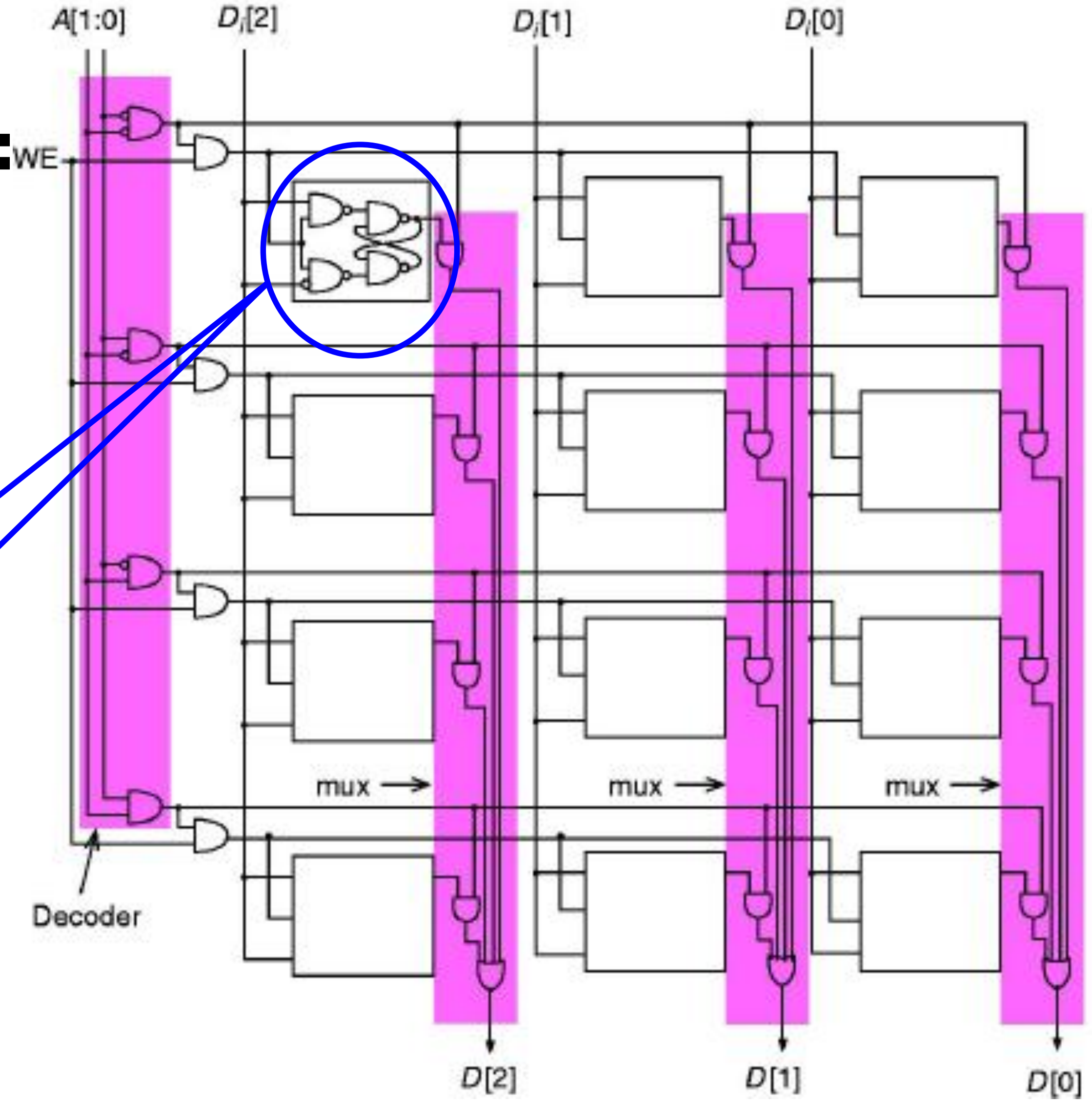
Memory Example

A 2^2 by 3 bits memory:

two address lines: A[1:0]

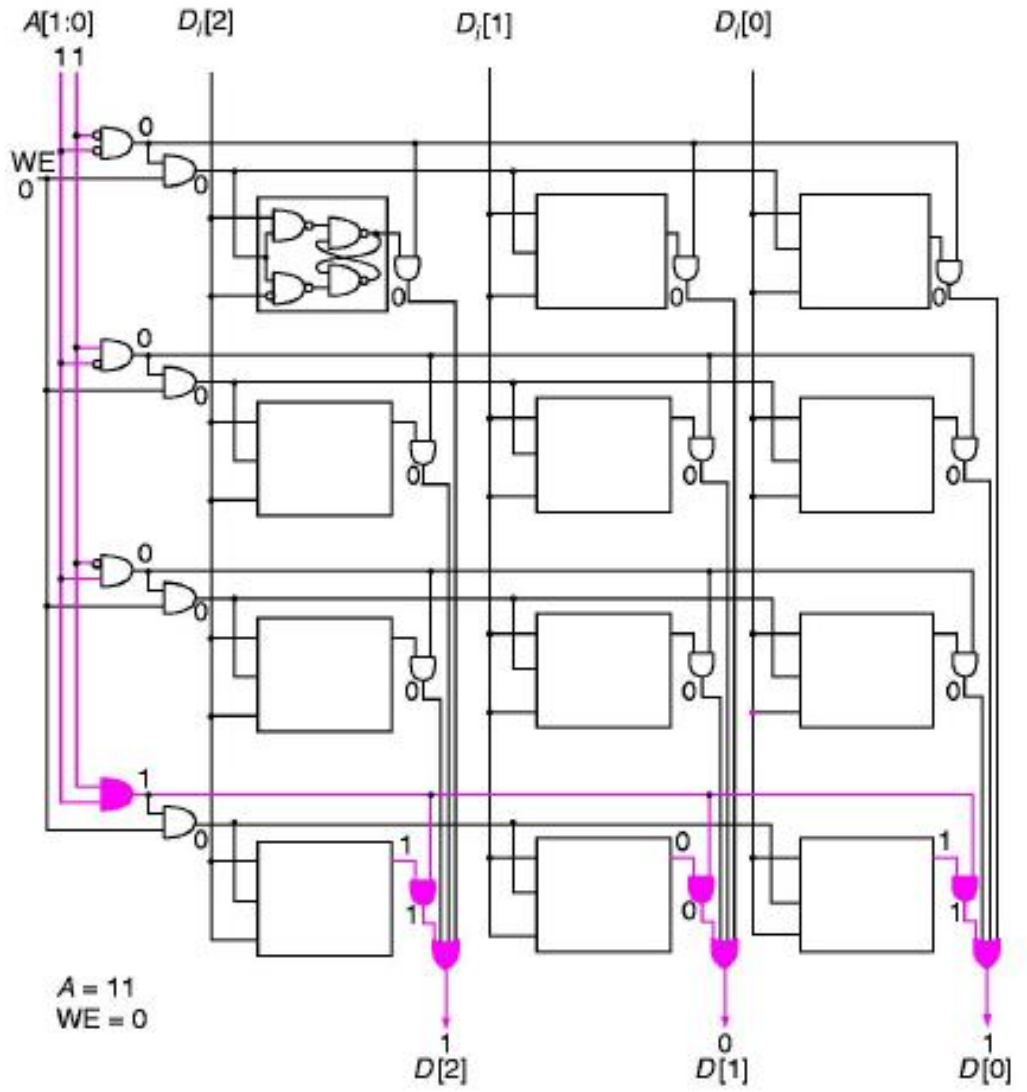
three data lines: D[2:0]

one control line: WE

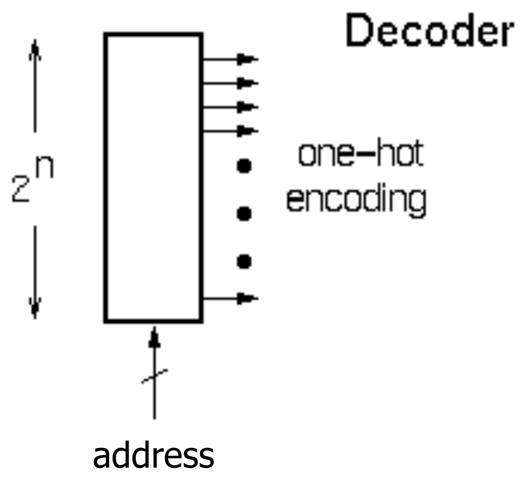


One gated D-latch

Reading a location in memory

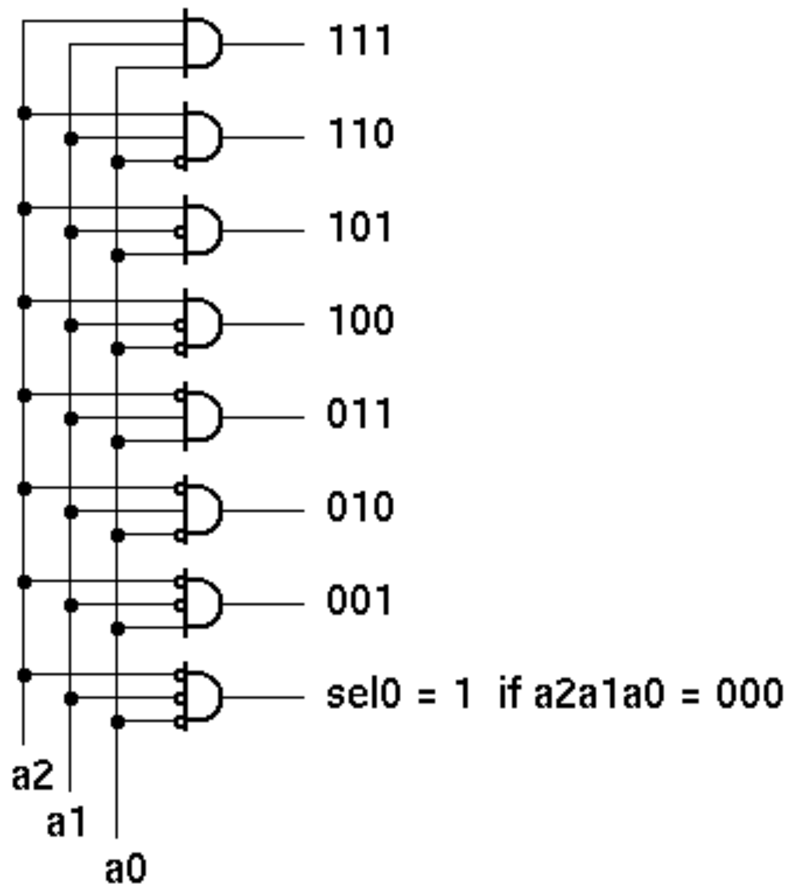


Address Decoding



sel1
sel0

addr	
000	00000001
001	00000010
010	00000100
011	00001000
100	00010000
101	00100000
110	01000000
111	10000000



Definitions

- **Bandwidth:**

  Total amount of data across out of a device or across an interface per unit time. (usually Bytes/sec)

- **Latency:**

  A measure of the time from a request for a data transfer until the data is received.

Memory Interfaces for Accessing Data

- **Asynchronous** (unclocked):

  A change in the address results in data appearing

- **Synchronous** (clocked):

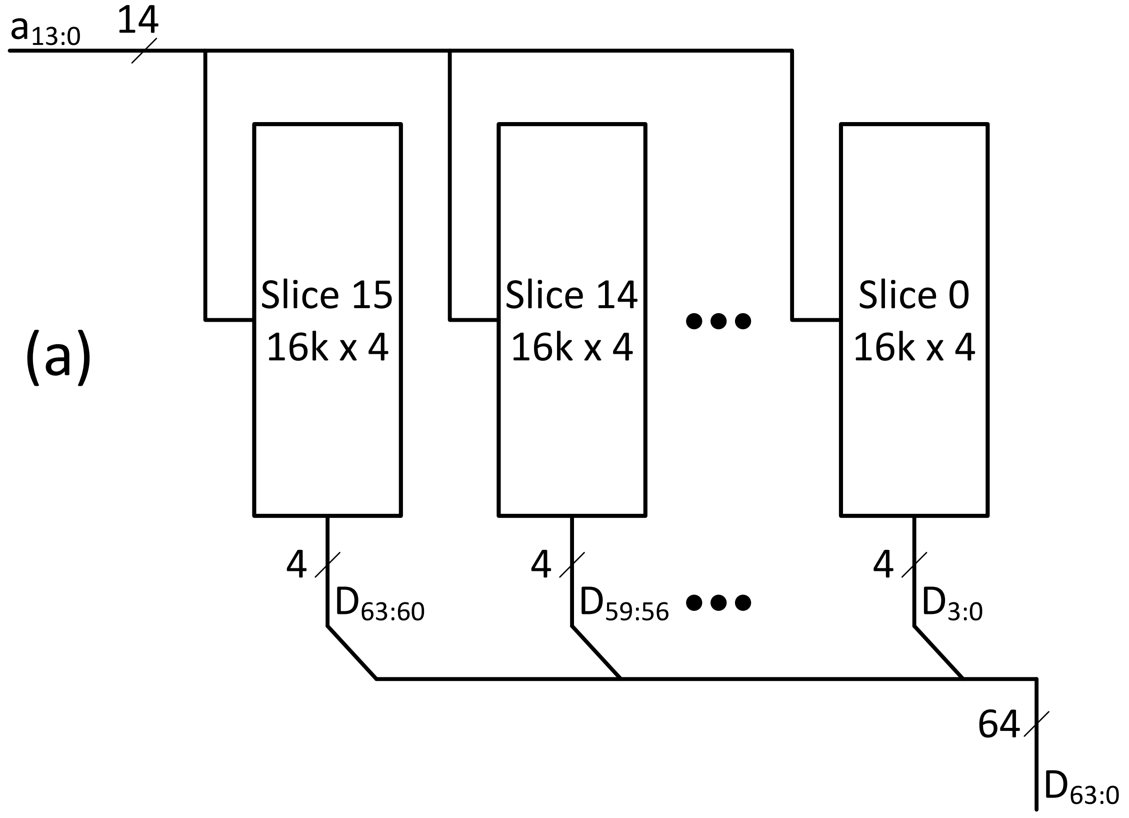
  A change in address, followed by an edge on CLK results in data appearing. Sometimes, multiple requests may be outstanding.

- **Volatile:**

  Looses its state when the power goes off. (the opposite: **non-volatile**)

What if you need more memory or more bandwidth than one primitive?

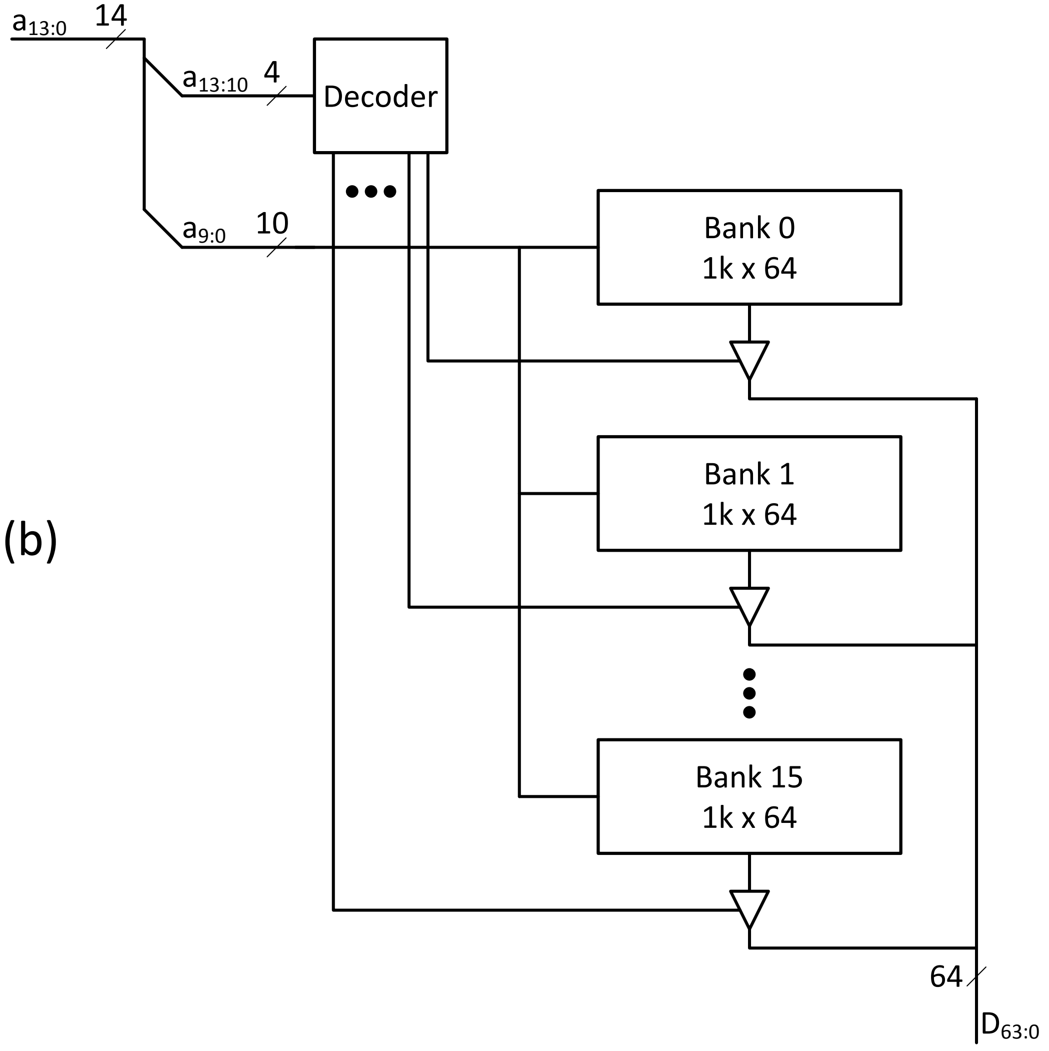
Bit-Slicing



$a_{13:0}$
14
(a)
Slice 15
16k x 4
Slice 14
16k x 4
•••
Slice 0
16k x 4
4
$D_{63:60}$
4
$D_{59:56}$
•••
4
$D_{3:0}$
64
$D_{63:0}$

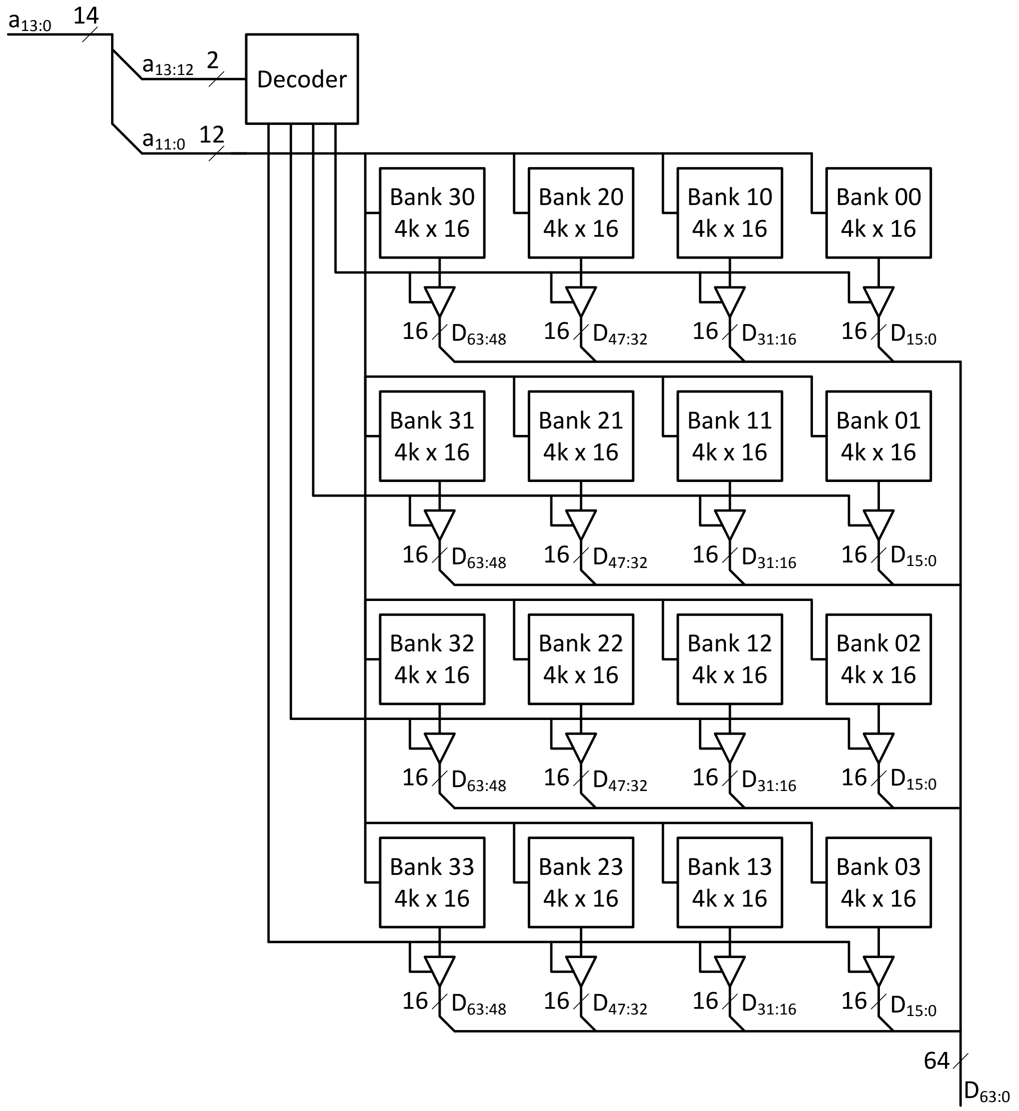
Banking

(b)

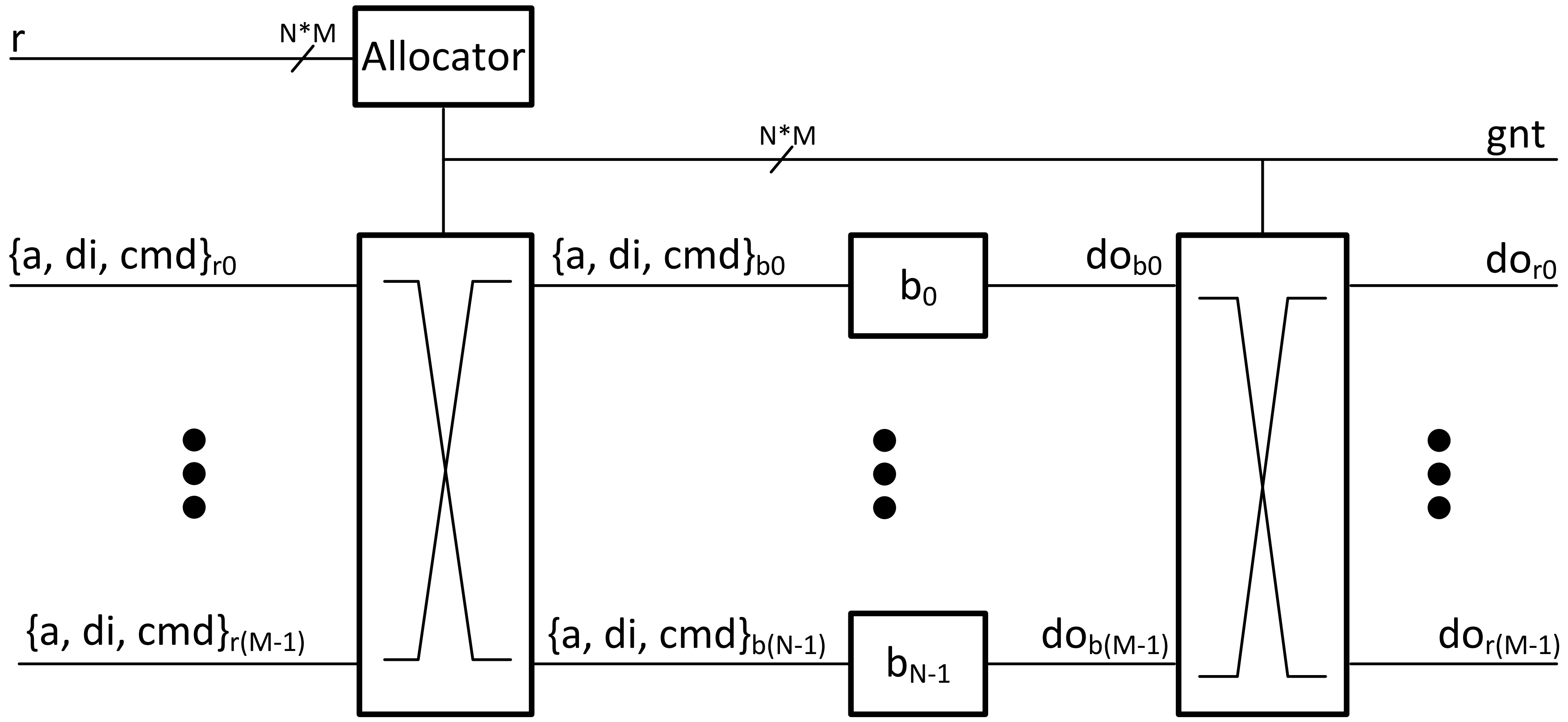


$a_{13:0}$
14
$a_{13:10}$
4
Decoder
$a_{9:0}$
10
Bank 0
1k x 64
Bank 1
1k x 64
Bank 15
1k x 64
64
$D_{63:0}$

Bit slicing & banking



Interleaving



Avoid head-of-line blocking

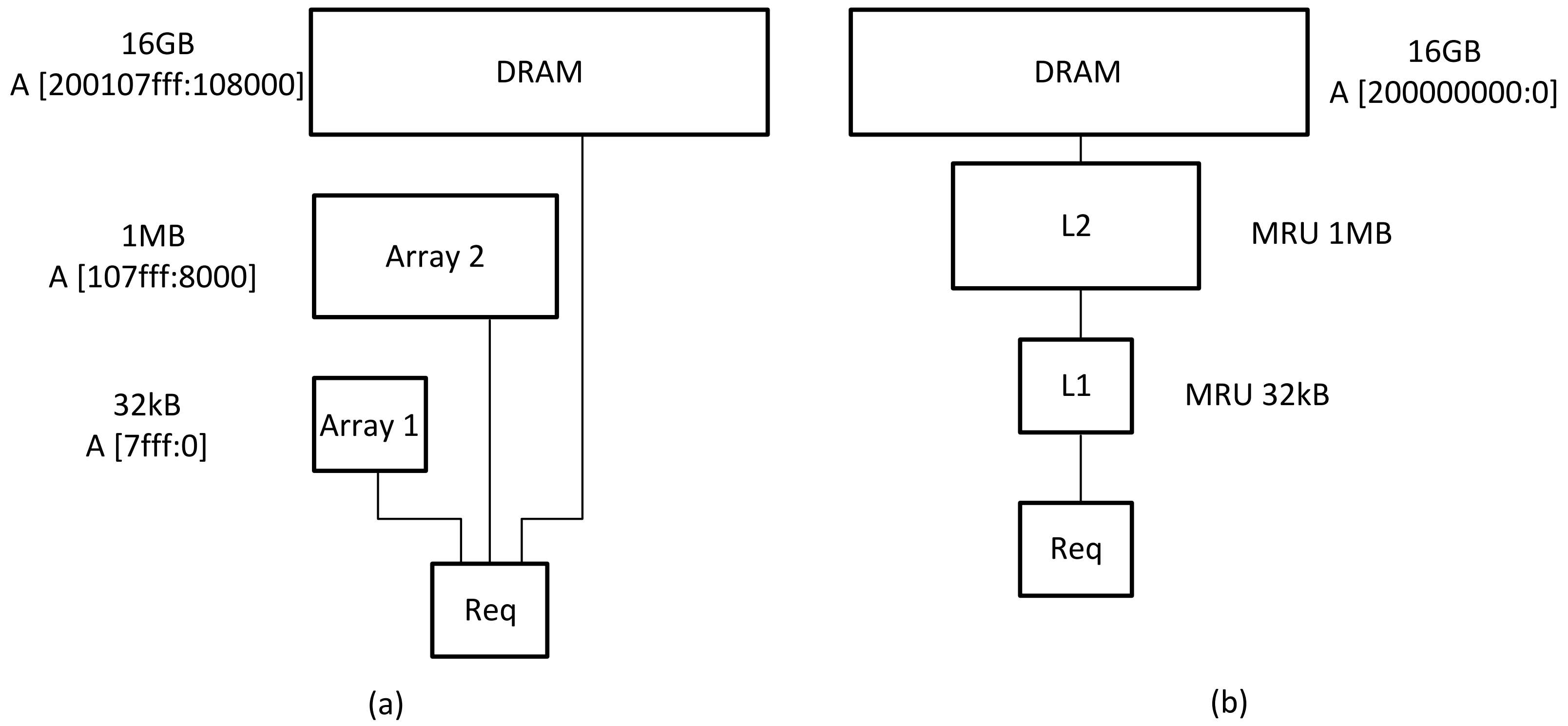
Time	Q0	Q1	Q2	Q3	G0	G1	G2	G3
1	0,1,2,3	0,1,2,3	0,1,2,3	0,1,2,3	Q0	–	–	–
2	1,2,3	0,1,2,3	0,1,2,3	0,1,2,3	Q1	Q0	–	–
3	2,3	1,2,3	0,1,2,3	0,1,2,3	Q2	Q1	Q0	–
4	3	2,3	1,2,3	0,1,2,3	Q3	Q2	Q1	Q0
5	–	3	2,3	1,2,3	–	Q3	Q2	Q1
6	–	–	3	2,3	–	–	Q3	Q2
7	–	–	–	3	–	–	–	Q3

Table 24.2: With head of line blocking, the memory system does not have full throughput. Requests that can be granted are stuck behind requests waiting for a over-subscribed resource.

Time	Q0	Q1	Q2	Q3	G0	G1	G2	G3
1	0,1,2,3	0,1,2,3	0,1,2,3	0,1,2,3	Q0	Q1	Q2	Q3
2	1,2,3	0,2,3	0,1,3	0,1,2	Q3	Q0	Q1	Q2
3	2,3	0,3	0,1	1,2	Q2	Q3	Q0	Q1
4	3	0	1	2	Q1	Q2	Q3	Q0

Table 24.3: When the arbiter is able to see requests beyond the head of the full, the memory system has no head of line blocking. It achieves full throughput.

Hierarchy


16GB
A [200107fff:108000]
DRAM
1MB
A [107fff:8000]
Array 2
32kB
A [7fff:0]
Array 1
Req
(a)
DRAM
16GB
A [200000000:0]
L2
MRU 1MB
L1
MRU 32kB
Req
(b)

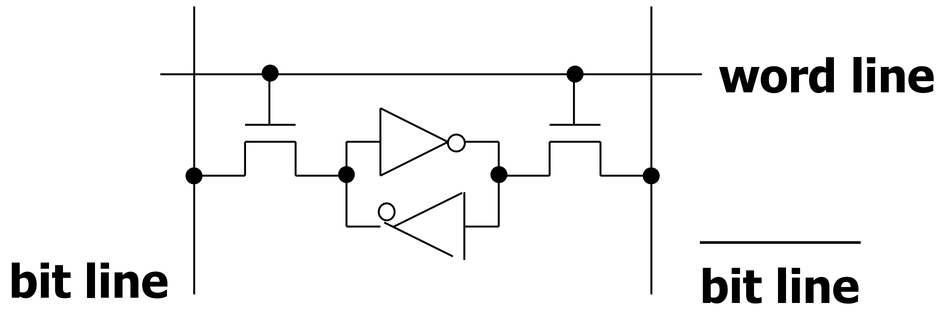
Example Memory Components:

- Volatile:
  - Random Access Memory (RAM):
    - DRAM "dynamic"
    - SRAM "static"

- Non-volatile:
  - Read Only Memory (ROM):
    - Mask ROM "mask programmable"
    - EPROM "electrically programmable"
    - EEPROM "erasable electrically programmable"
    - FLASH memory - similar to EEPROM with programmer integrated on chip

Volatile Memory Comparison

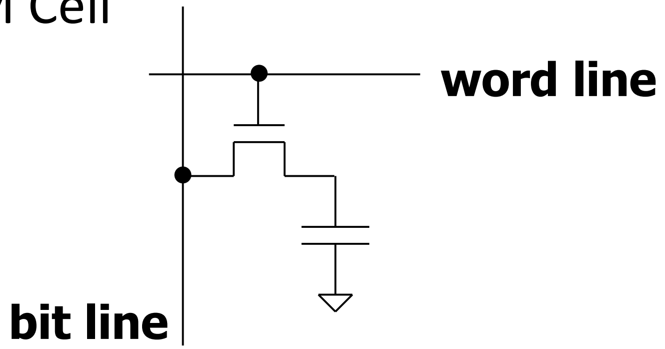
- SRAM Cell



- Larger cell ⇒ lower density, higher cost/bit
- No refresh required
- Simple read ⇒ faster access
- Standard IC process ⇒ natural for integration with logic

- **Read:** Pre-charge bit-lines, Raise wordline, cell pulls one bit line low, sense the difference
- **Write:** Drive data onto bit-lines (one 0, one 1), Raise wordline

- DRAM Cell



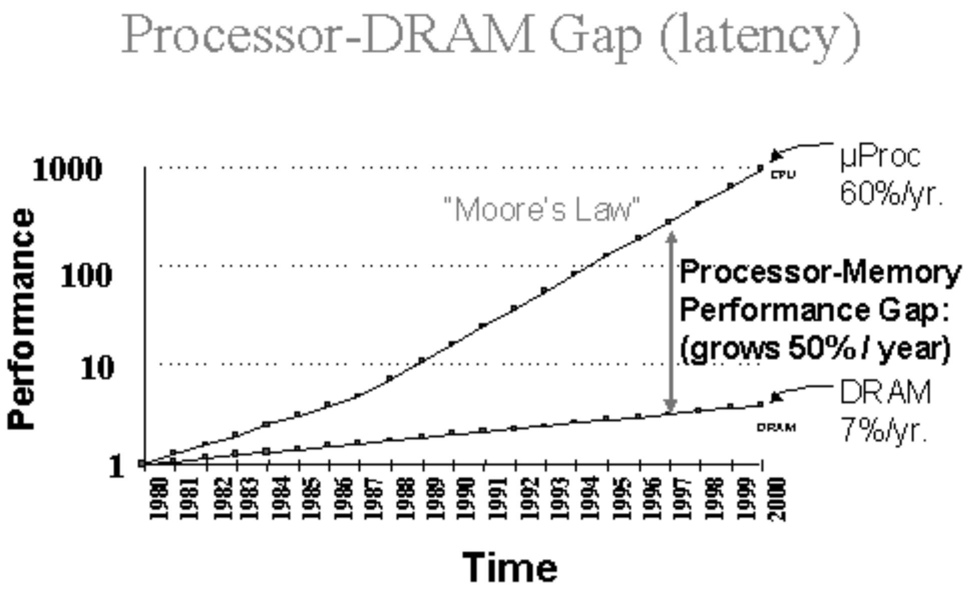
- Smaller cell ⇒ higher density, lower cost/bit
- Needs periodic refresh, and refresh after read
- Complex read ⇒ longer access time
- Special IC process ⇒ difficult to integrate with logic circuits

- **Read:** Pre-charge bit-lines with Vdd/2, Raise wordline, bit-line gets the read value, value sensed, and written back to the cell
- **Write:** put the value to write in the bit-line, Raise wordline
- **Refresh:** a dummy read to every cell.

In Desktop Computer Systems:

- **SRAM** (lower density, higher speed) used in CPU register file, on- and off-chip caches.
- **DRAM** (higher density, lower speed) used in main memory

Processor-DRAM Gap (latency)



Closing the GAP: Innovation targeted towards higher bandwidth for memory systems:

SDRAM - synchronous DRAM

RDRAM - Rambus DRAM

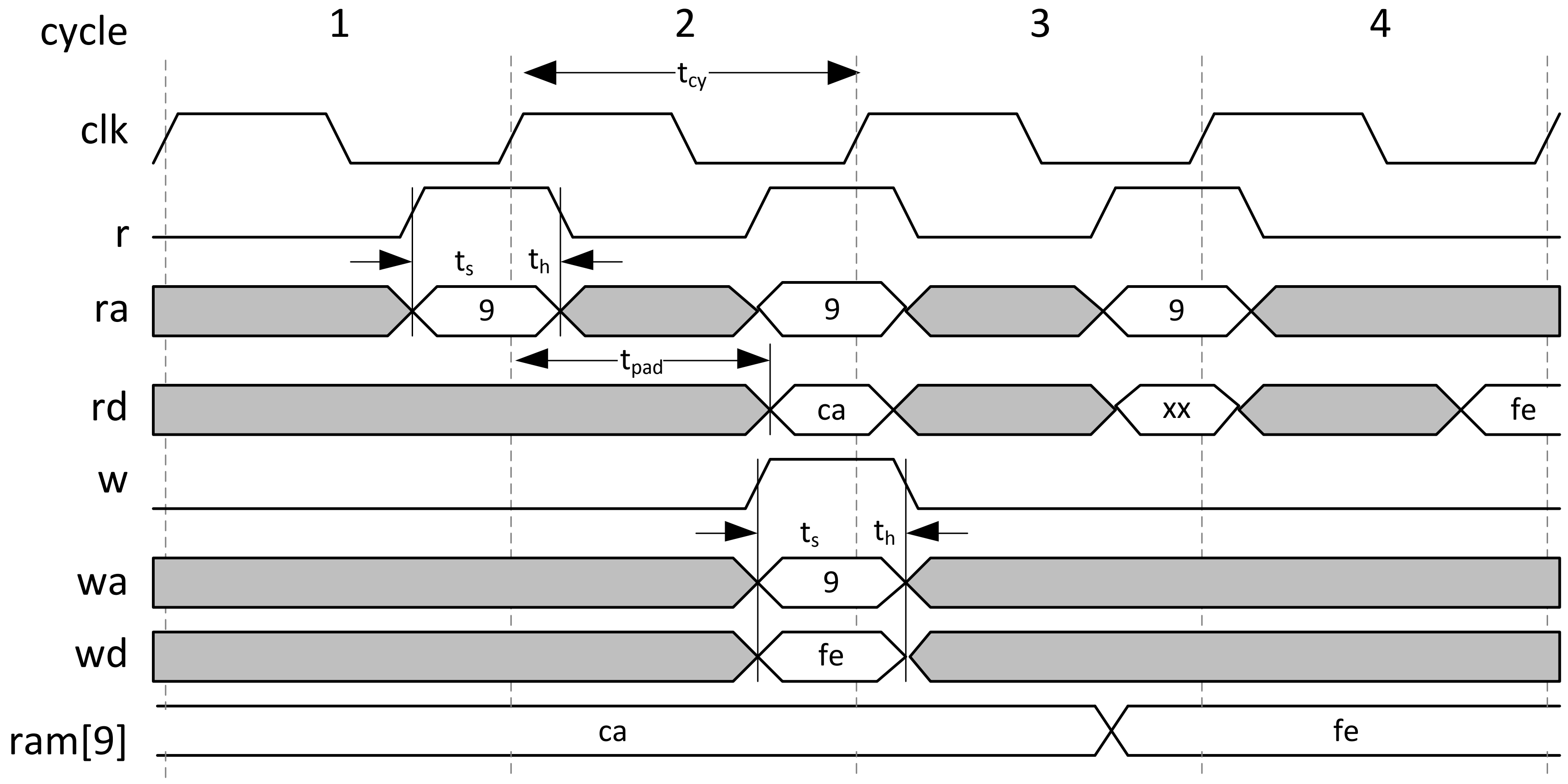
EDORAM - extended data out SRAM

3D-stacked DRAM

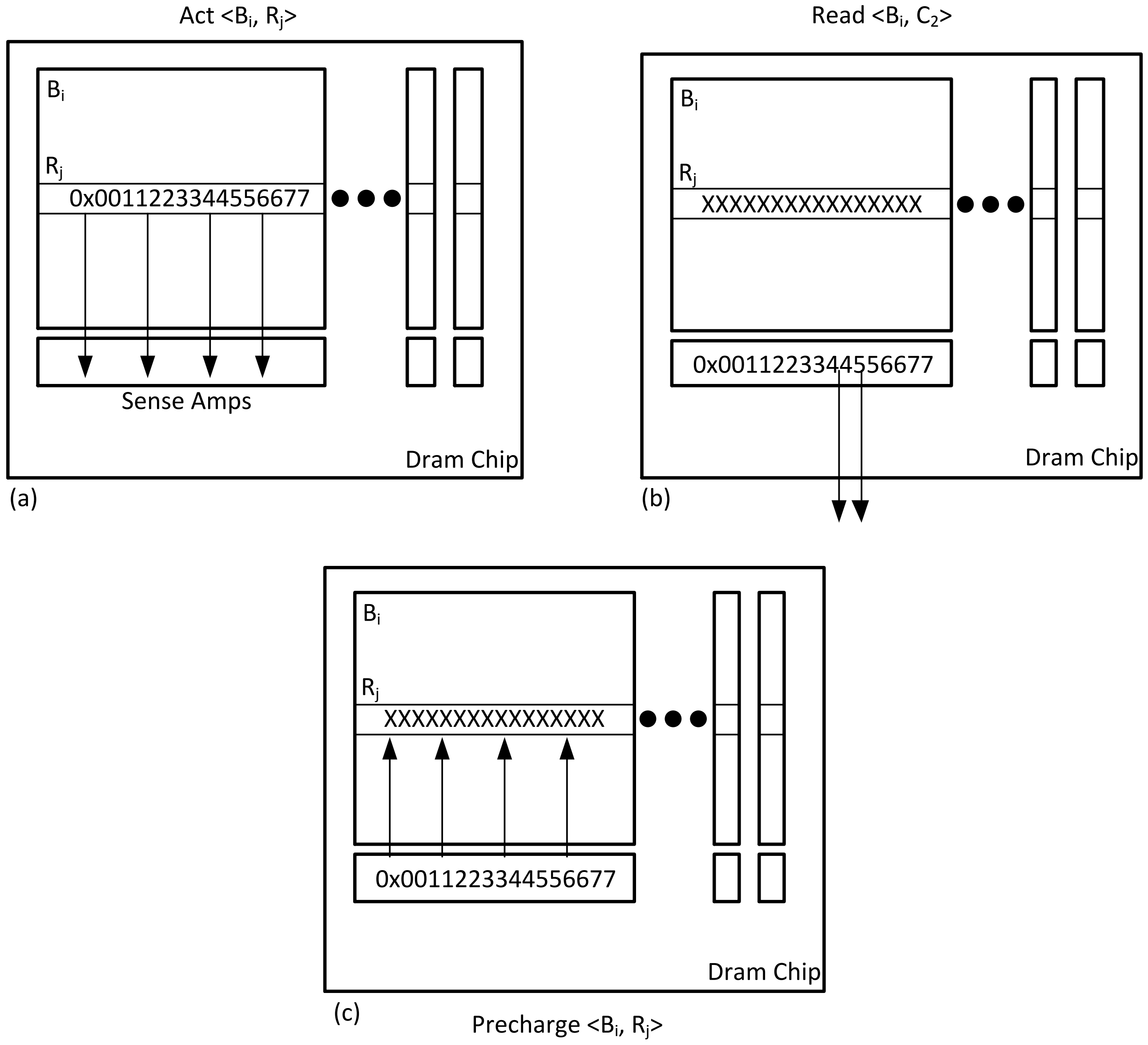
hyper-page mode DRAM video RAM

multibank DRAM

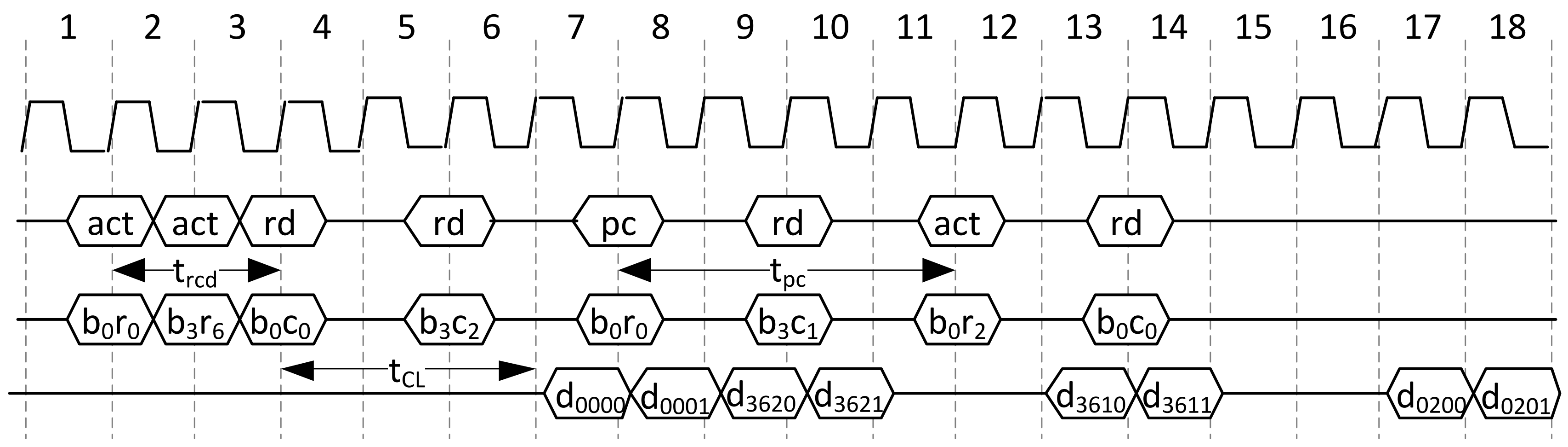
SRAM Primitive



DRAM Operation

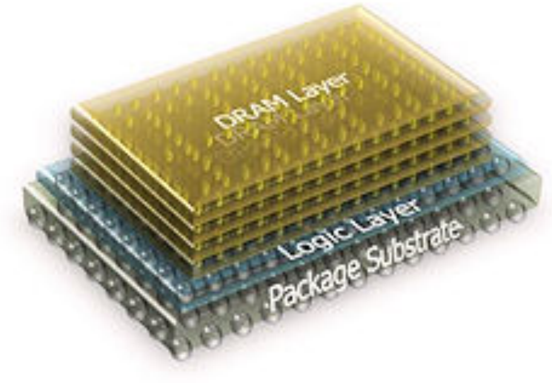


DRAM Primitive



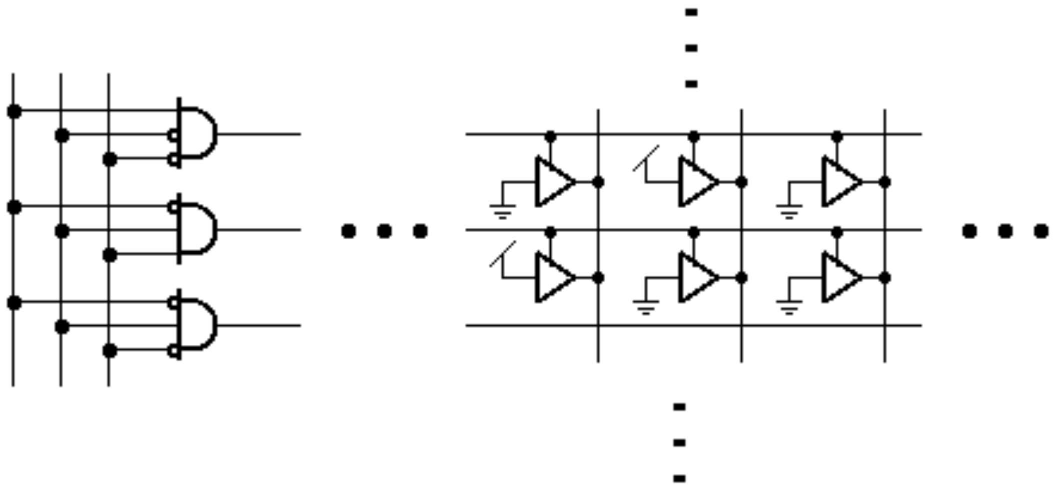
Important DRAM Examples:

- SDRAM - synchronous DRAM
  - clocked interface
  - uses dual banks internally. Start access in one bank then next, then receive data from first then second.
- DDR - Double data rate SDRAM
  - Uses both rising (positive edge) and falling (negative) edge of clock for data transfer. (typical 100MHz clock with 200 MHz transfer).
- RDRAM - Rambus DRAM
  - Entire data blocks are accessed and transferred out on a high-speed bus-like interface (500 MB/s, 1.6 GB/s)
  - Tricky system level design. More expensive memory chips.
- 3D stacked DRAM (e.g. Micron's HMC)
  - DRAM dies stacked one above the other
  - Connected through through-silicon-vias (TSV)
  - Logic layer for memory control
  - 160 GB/s



Read Only Memory (ROM)

- Functional Equivalence:



- Of course, full tri-state buffers are not needed at each cell point.
- Single transistors are used to implement zero cells. Logic one's are derived through *precharging* or bit-line *pullup transistor*.

Non-volatile Memory

Used to hold fixed code (ex. BIOS), tables of data (ex. FSM next state/output logic), slowly changing values (date/time on computer)

- Mask ROM
  - Used with logic circuits for tables etc.
  - Contents fixed at IC fab time (truly write once!)
- EPROM (erasable programmable)

  & FLASH
  - requires special IC process

    (floating gate technology)

Cell Operation: Programming

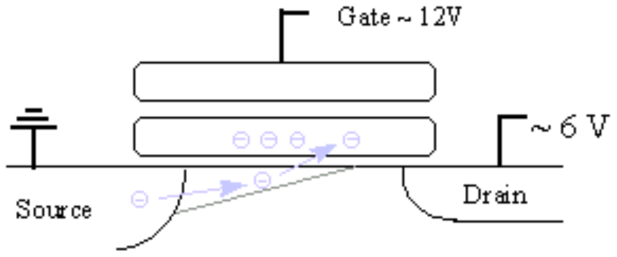


Figure 2: Cell bias conditions during programming

  - writing is slower than RAM. EPROM uses special programming system to provide special voltages and timing.
  - reading can be made fairly fast.
  - rewriting is very slow.
    - erasure is first required , EPROM - UV light exposure

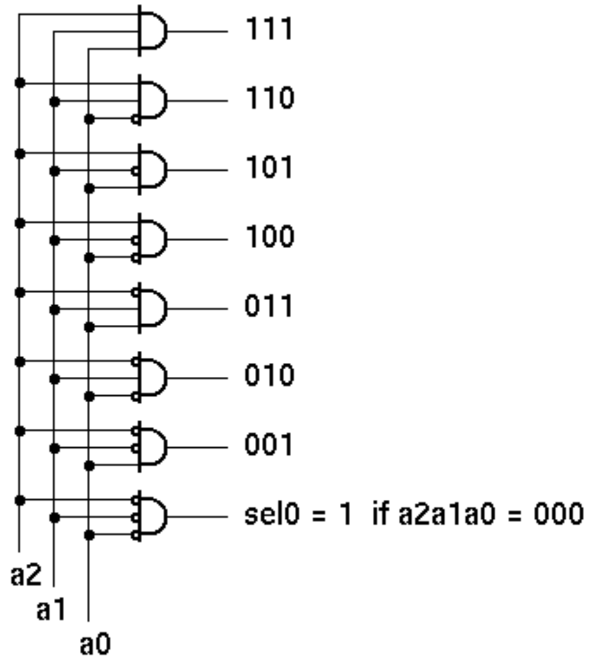
FLASH Memory

- Electrically erasable
- In system programmability and erasability (no special system or voltages needed)
- On-chip circuitry (FSM) to control erasure and programming (writing)
- Erasure happens in variable sized "sectors" in a flash (16K - 64K Bytes)

See: http://developer.intel.com/design/flash/
for product descriptions, etc.

Relation between Memory & Combinational logic

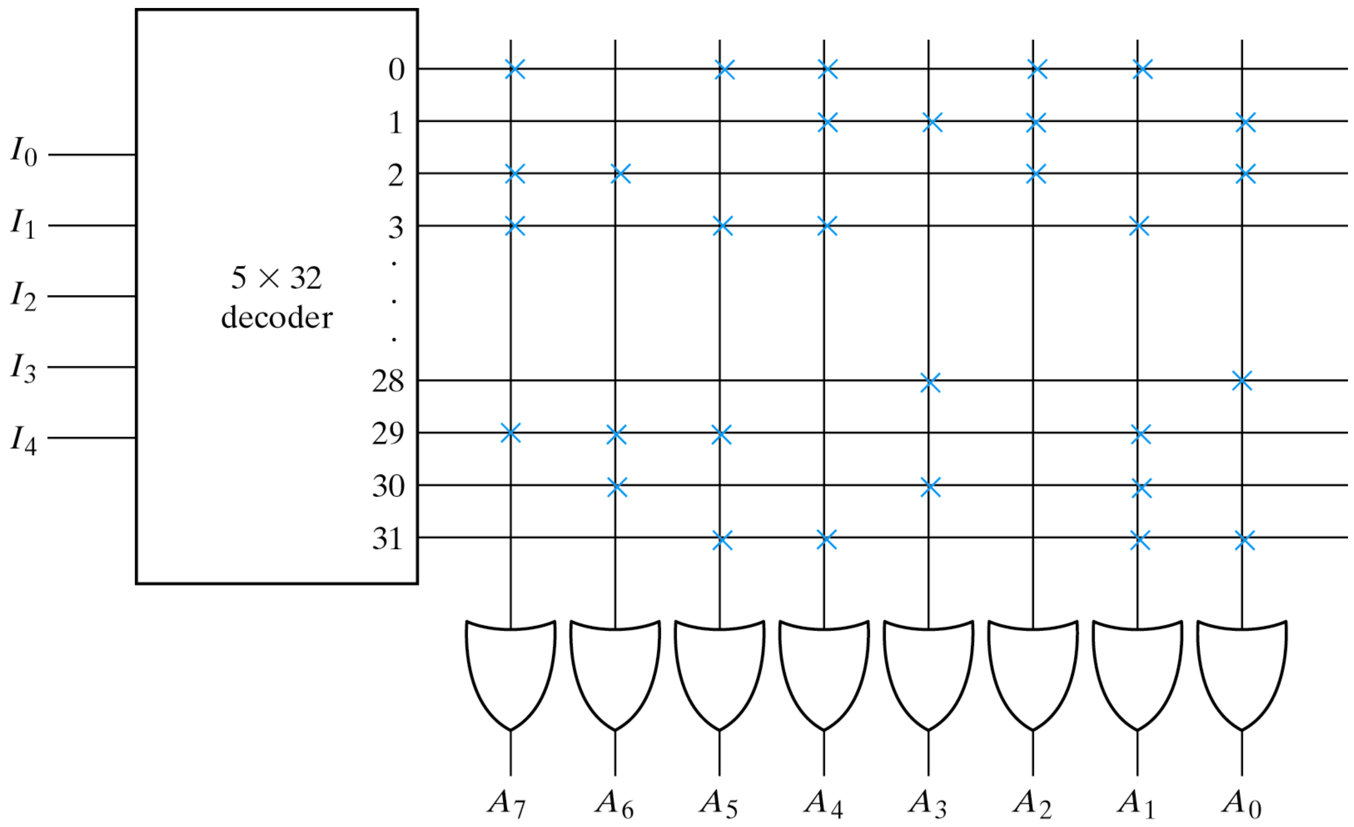
- Memory blocks can be (and often are) used to implement combinational logic functions:
- Examples:
  - LUTs in FPGAs
  - 1Mbit x 8 EPROM can implement 8 independent functions each of $\log_2(1M)=20$ inputs.
- The *decoder part* of a memory block can be considered a "minterm generator".
- The *cell array part* of a memory block can be considered an OR function over a subset of rows.

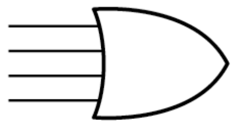


The combination gives us a way to implement logic functions directly in sum of products form.

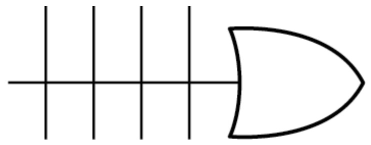
Several variations on this theme exist in a set of devices called Programmable logic devices (PLDs)

A ROM as AND/OR Logic Device



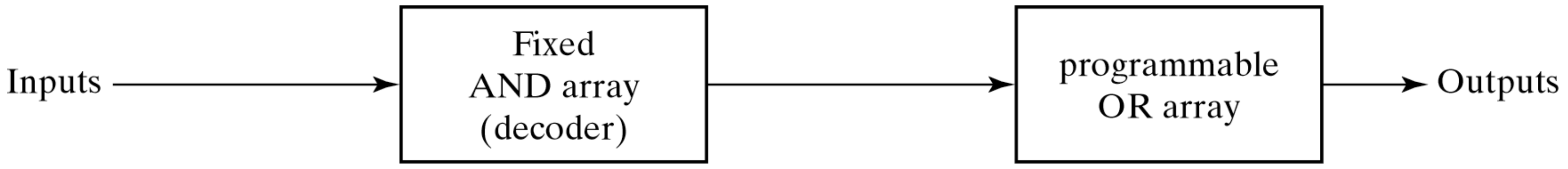


(a) Conventional symbol

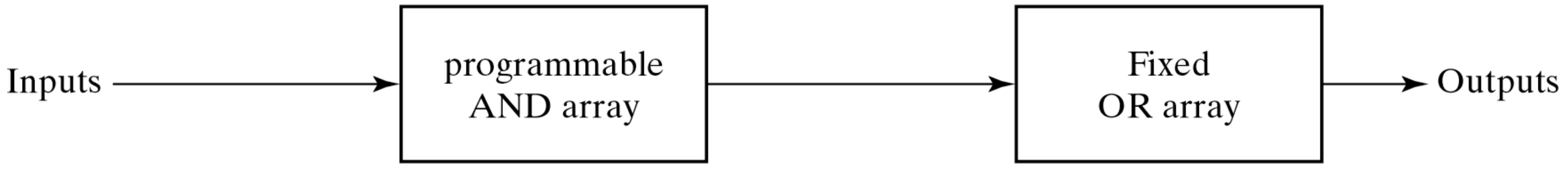


(b) Array logic symbol

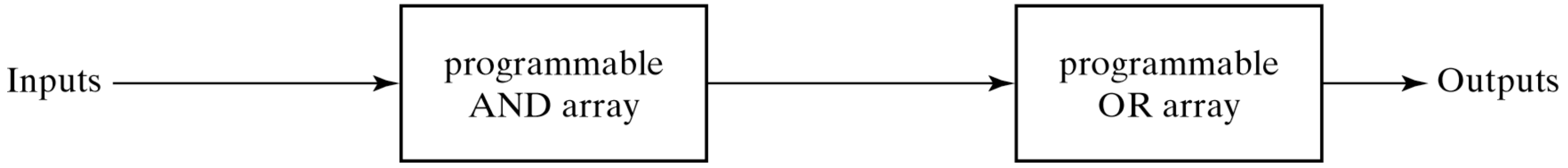
PLD Summary



(a) Programmable read-only memory (PROM)



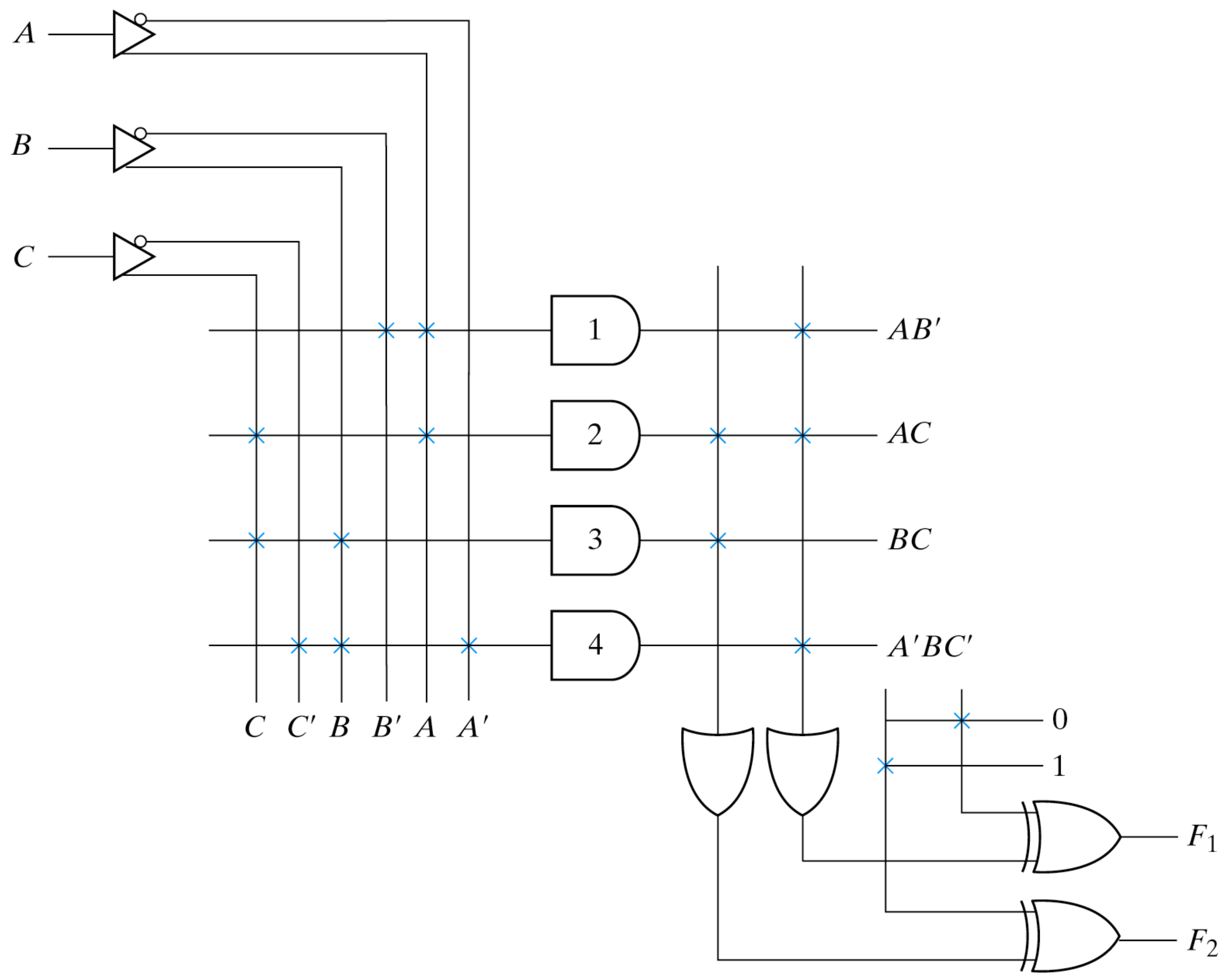
(b) Programmable array logic (PAL)



(c) Programmable logic array (PLA)

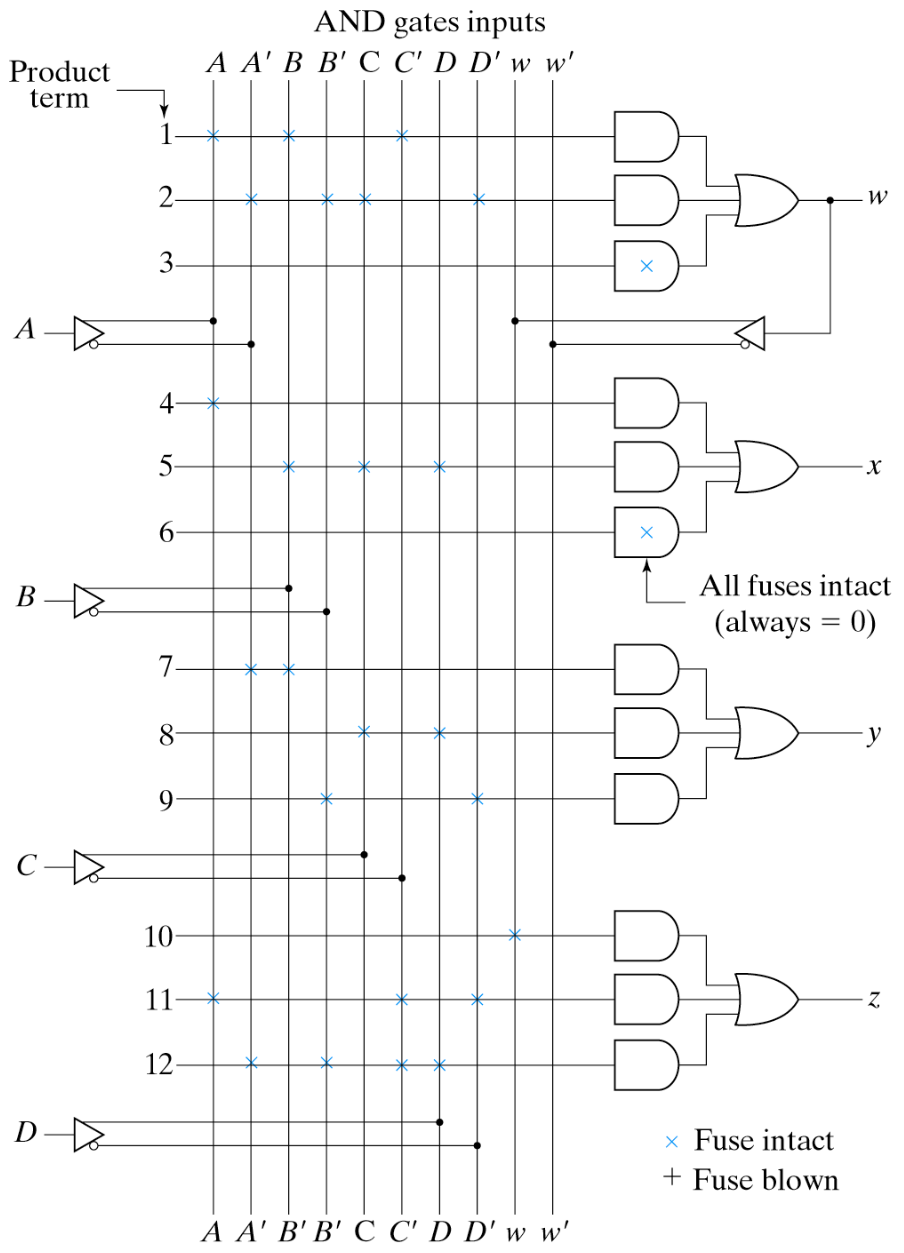
Basic Configuration of Three PLDs

PLA Example



PLA with 3 Inputs, 4 Product Terms, and 2 Outputs

PAL Example



Fuse Map for PAL as Specified in Table 7-6

Memory Blocks in FPGAs

- LUTs can implement either a logic gate or a small RAM blocks:
  - 4-LUT is a 16x1 memory or a 4 input programmable gate

- Newer FPGA families include additional on chip RAM blocks (usually dual ported)
  - Called "block-rams" or "BRAMs" in Xilinx Virtex series, e.g. 16k x 1bit

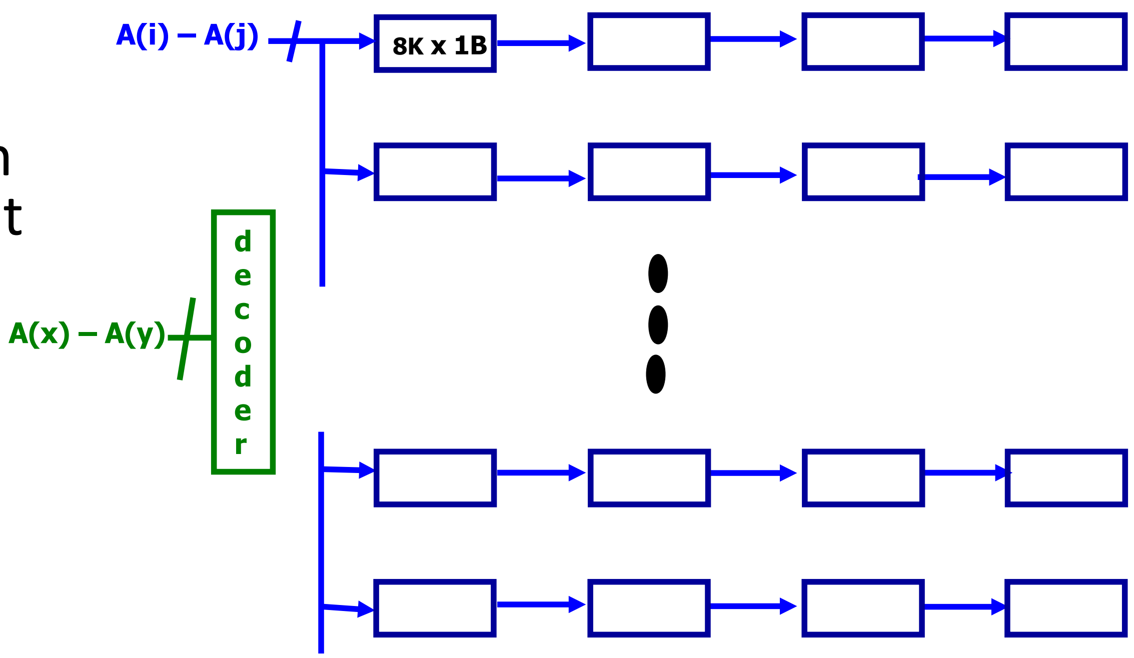
Quiz 11-2

http://m.socrative.com/student/#joinRoom

room number: 713113

- Q1: The non-volatile memory looses its contents at power off
- Q2: DRAM needs refreshing of its contents in order to keep them, SRAM does not
- Q3: how many bits of address requires a memory with 4096 entries?
- Q4: Use memory blocks of 8K x 1 byte to build a memory that is 64K words, with each location one word of 32 bits. What are the address lines of the memory?
  - Find i, j, x,y

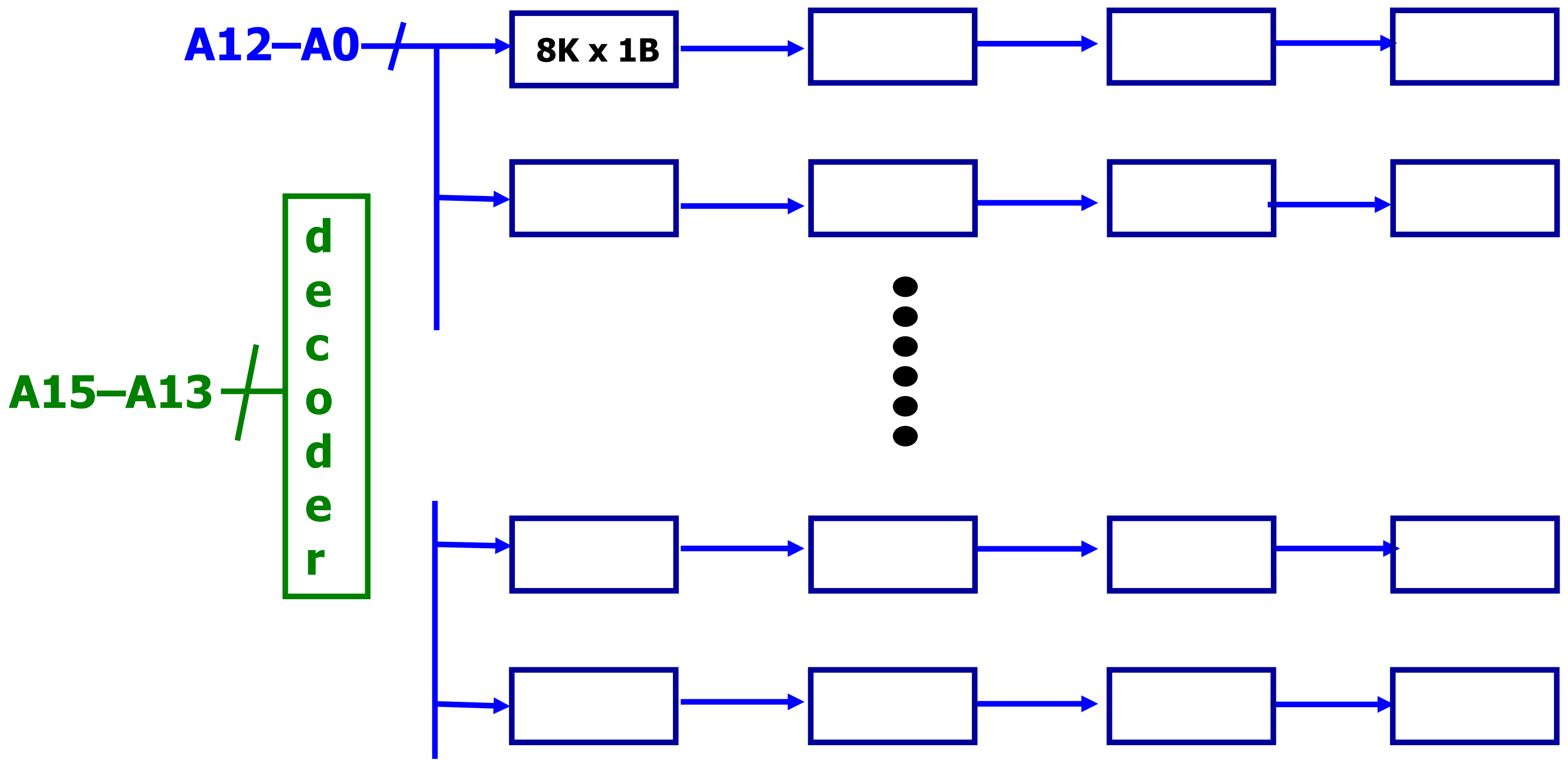
  separate with space, e.g. 1 2 3 4



Memory One Word Wide

Use the previous memory block of 8K x 1 byte to build a memory that is 64K words, with each location one word of 32 bits.

- what are the address lines if the memory is **word addressed**? or byte addressed?



Summary of Lecture 14

- Interconnect
  - Allow any pair of clients to communicate
  - Bus, Crossbar, Network

- Memory
  - Standard memory organization
  - Capacity, bandwidth, latency, and granularity
  - RAM, ROM
  - SRAM and DRAM primitives
  - Bit-slice, bank, interleave to combine primitives
  - Hierarchy
  - Memory used for combinational logic

- Reading
  - Book chapters 24, 25

- Next Lecture 15:
  - Test and verification