BROWN
School of Engineering

DIGITAL ELECTRONICS SYSTEM DESIGN

FALL 2019

PROF. IRIS BAHAR

NOVEMBER 25, 2019

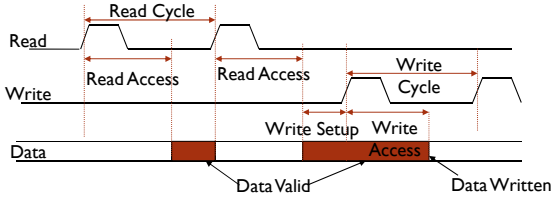
LECTURE 22: MEMORY DESIGN

LAB HOURS THIS WEEK

- Eren will not be holding lab hours this Tuesday evening.
- Monica still has hours on Wednesday from 11am-1pm (unless no one is planning to show up…)

MEMORY DEFINITIONS

- Size (Kbytes, Mbytes, Gbytes, Tbytes)
- Timing parameters
  - Read Access – delay between read request and the data available
  - Write Access – delay between write request and writing data to memory
  - (Read or Write) Cycle – min. time required between read/writes



MEMORY DEFINITIONS

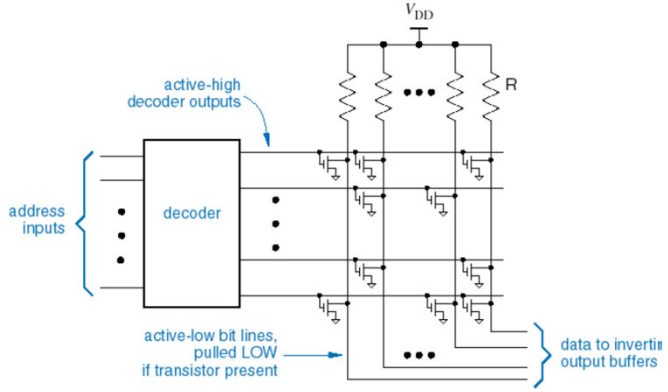
- Function – functionality, nature of the storage mechanism
  - static and dynamic; volatile and nonvolatile
- Access pattern – random, serial, content addressable
- Input-output architecture – number of data input and output ports (multi-ported memories)
- Application – embedded, secondary, tertiary

MEMORY DEFINITIONS

RWM		NVRWM	ROM
Random Access	Non-Random Access	EPROM	Mask-programmed
SRAM (cache, register file)	FIFO, LIFO	EEPROM	
DRAM (main memory)	Shift Register	FLASH	Electrically-programmed (PROM)
	CAM		

- Read-Only Memories (ROM)
  - Truly read-only
    - Written in the factory, and never written after installation
  - Mostly read and rarely written
    - Much faster to read than write
  - Non-volatile, e.g., flash memory
- Random Access Memory (RAM)
  - Read and write any location at similar speeds
  - Volatile: loses contents when powered off
  - SRAM, DRAM

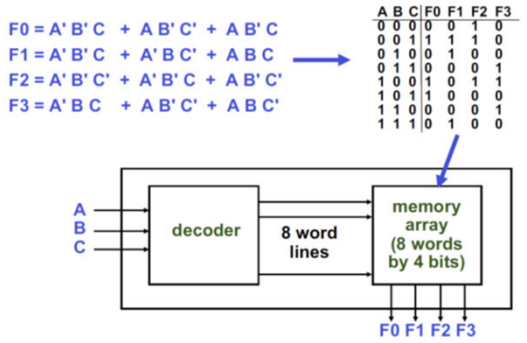
EXAMPLE ROM IMPLEMENTATION



APPLICATIONS OF ROM

- Program storage
  - e.g., Boot code for personal computers, application storage for embedded systems
- Data storage
  - e.g., Configuration information, music players
- Combinational logic functions
  - Lookup tables
    - Address inputs = function inputs
    - Data outputs = function outputs

USING ROMS FOR COMBINATIONAL LOGIC

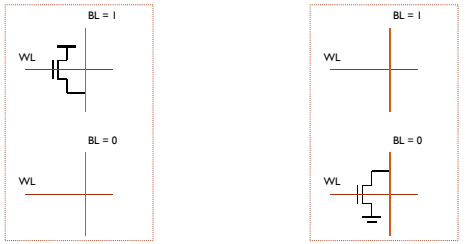


ROM AS A "RANDOM-LOGIC" CIRCUIT

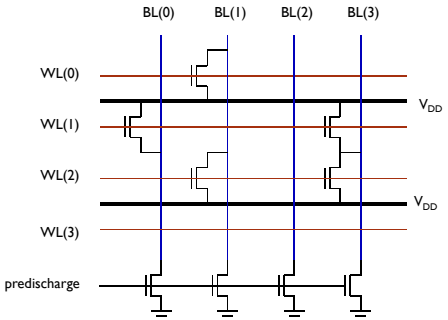
- ROM advantages
  - Design time is short
  - Can implement any function of n inputs
- ROM disadvantages
  - Size doubles for each additional input
  - Cannot exploit logic minimization (e.g., don't cares)

READ ONLY MEMORIES (ROMS)

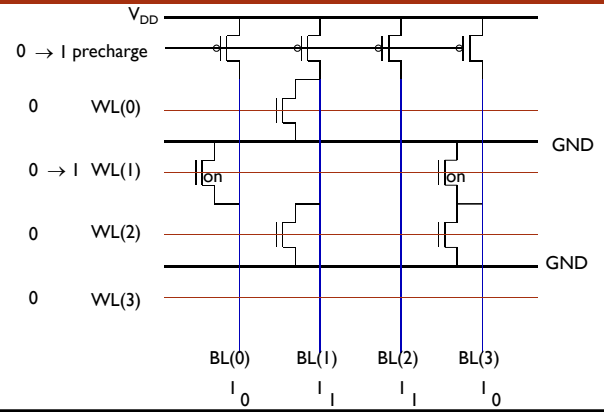
- A memory that can only be read and never altered
  - Programs for fixed applications that, once developed and debugged, never need to be changed (only read)
  - Fixing the contents at manufacturing time leads to small and fast implementations.



MOS OR ROM CELL ARRAY

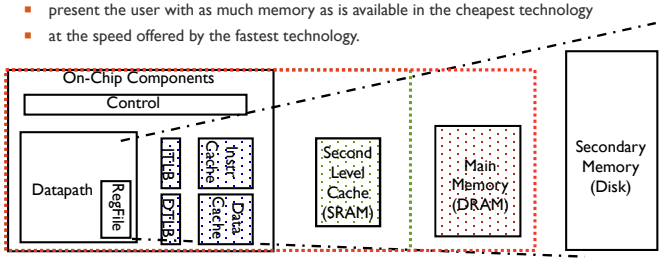


PRECHARGED MOS NOR ROM



A TYPICAL MEMORY HIERARCHY

- By taking advantage of the principle of locality, we can:
  - present the user with as much memory as is available in the cheapest technology
  - at the speed offered by the fastest technology.

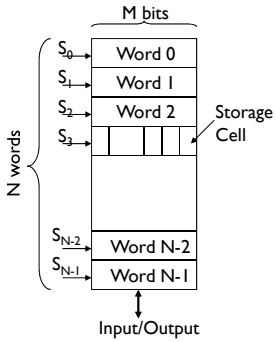


Speed (ns):	.1's	1's	10's	100's	1,000's
Size (bytes):	100's	KB's	10KB's	MB's	TB's
Cost:	highest				lowest

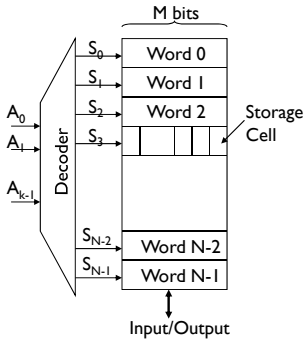
READ/WRITE MEMORIES (RAMS)

- Static – SRAM
  - data is stored as long as voltage supply is enabled
  - large cells (6 FETs/cell) → fewer bits/chip
  - fast → used where speed is important (e.g., caches)
- Dynamic – DRAM
  - periodic refresh required (every 1 to 4 ms)
  - small cells (1 to 3 FETs/cell) → more bits/chip
  - slower → used for main memories

1-DIMENSIONAL MEMORY ARCHITECTURE

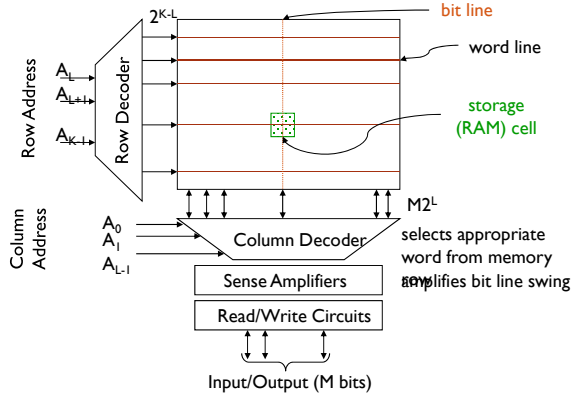


N words → N select signals

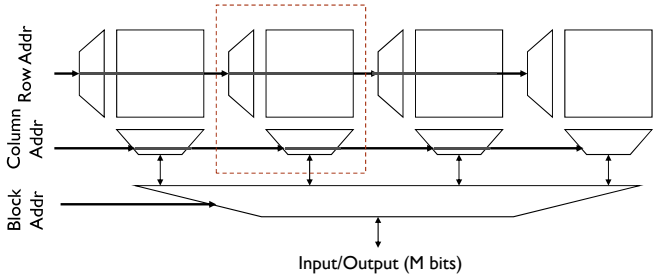


Decoder reduces # of inputs
$K = \log_2 N$

2D MEMORY ARCHITECTURE



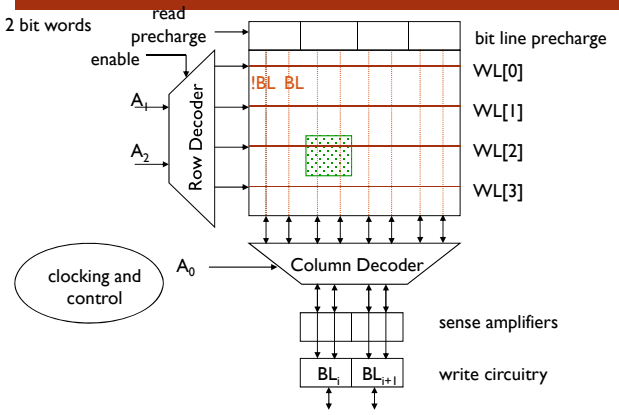
3D (I.E., BANKED) MEMORY ARCHITECTURE



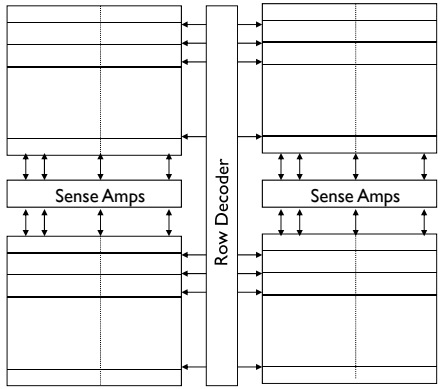
Advantages:

1. Shorter word and bit lines so faster access
2. Block addr activates only 1 block saving power

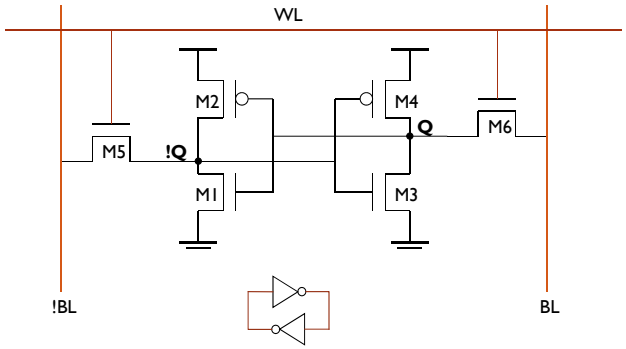
4X4 SRAM MEMORY



2D MEMORY CONFIGURATION



6-TRANSISTOR SRAM CELL



SRAM WRITE

- Drive on bit line high, the other low (depending on the desired write value)
- Then enable the word line (WL)
- Bit lines overpower cell with new value