Computer Architecture: Fundamentals, Tradeoffs, Challenges

Chapter 6: Physical Memory

Yale Patt

The University of Texas at Austin

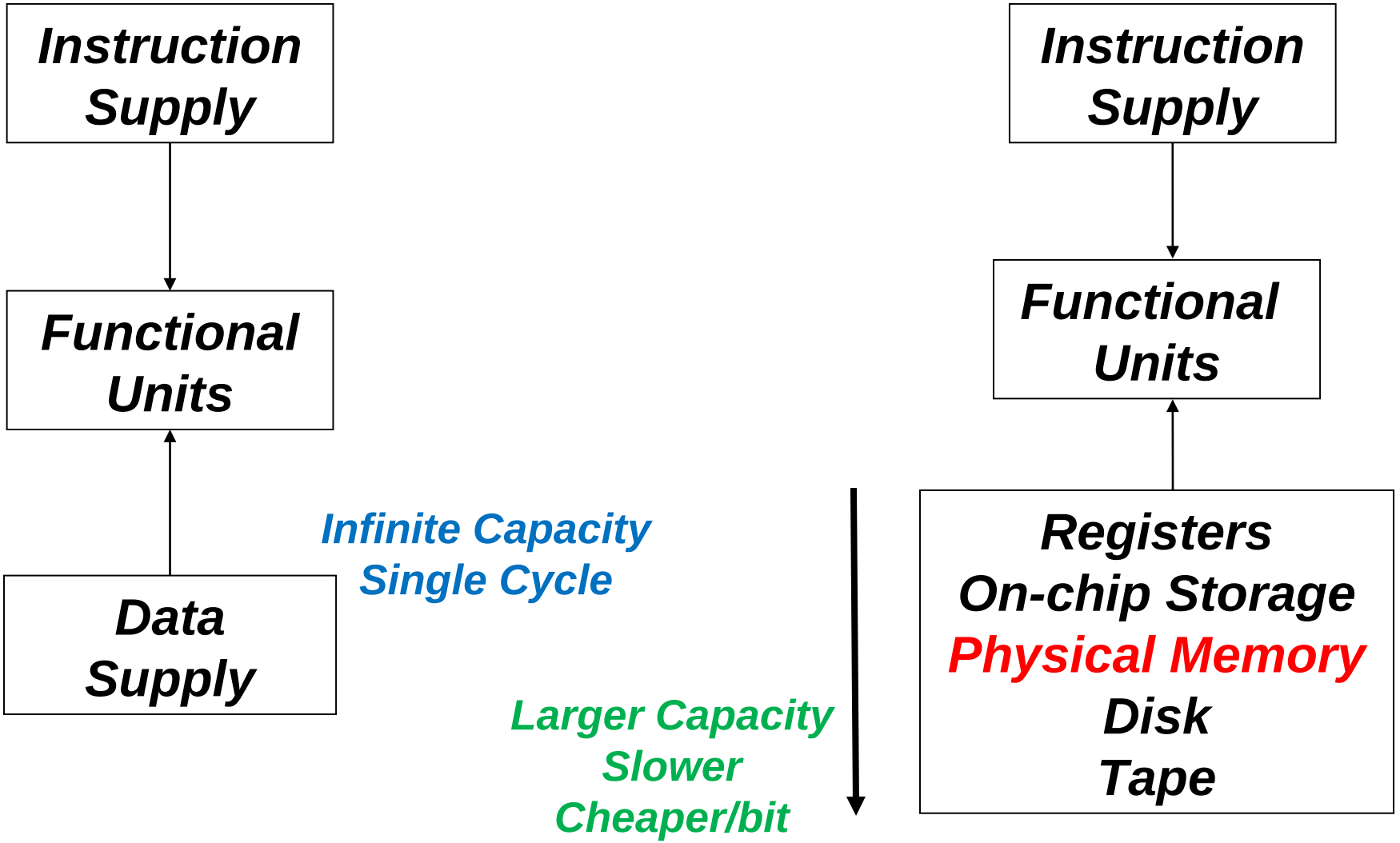
Austin, Texas

Spring, 2023

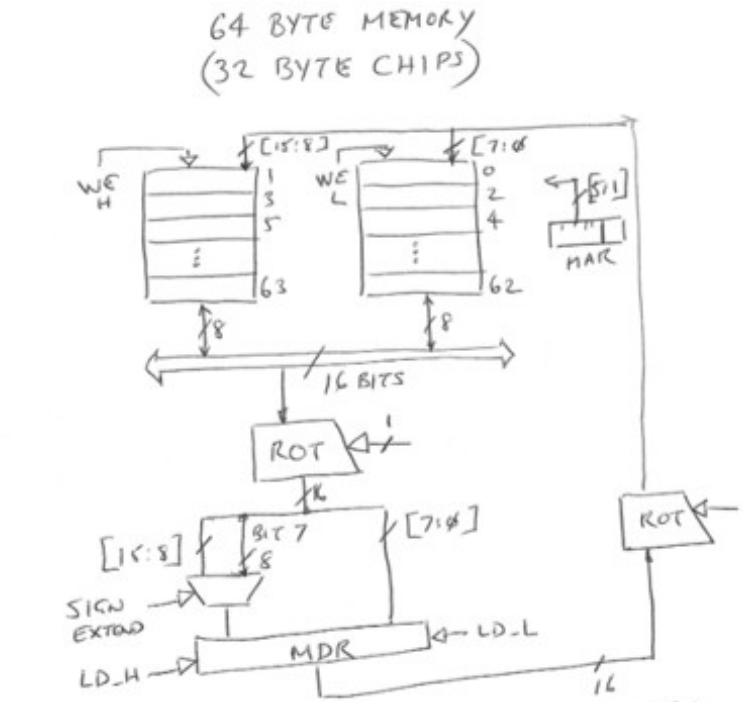
Outline

- ***The Storage hierarchy***
  - ***Structures: Registers, L1/L2/L3…Cache, Memory, Disk, Tape***
  - ***Access: RAM, DASD, Sequential, CAM***
- ***Two important concepts***
  - ***Interleaving***
  - ***Unaligned Access***
- ***Device Technology: Mag. Cores, SRAM, DRAM, NVM***
  - ***The DRAM chip***
    - ***Multiple Banks***
    - ***Row Buffer***
- ***The Memory Controller***
- ***Error Detection, Correction***

The Storage Hierarchy

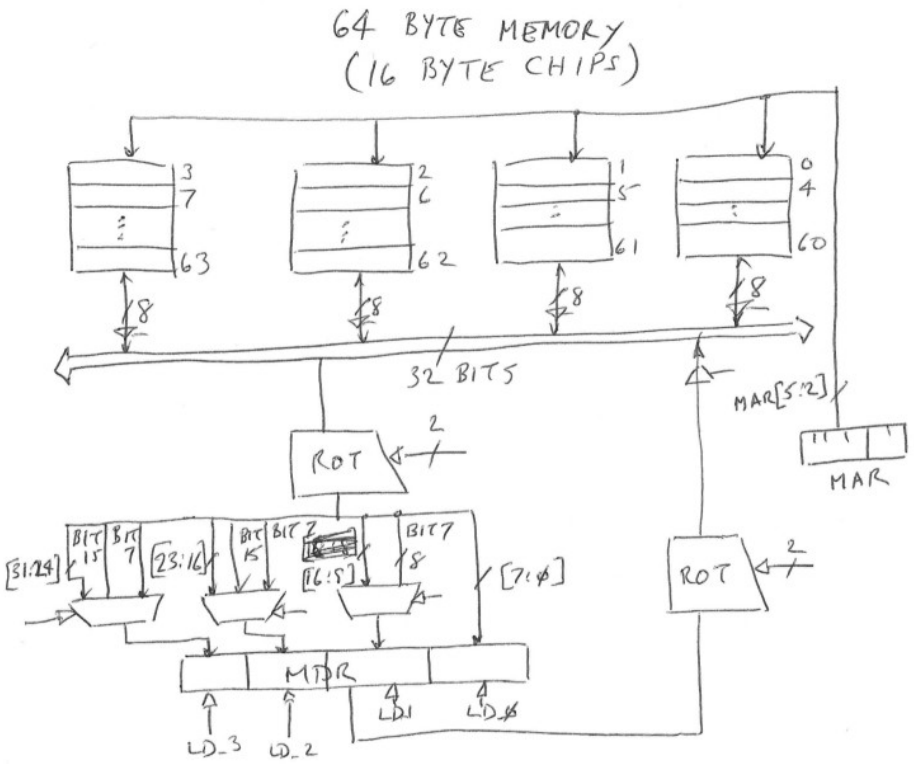


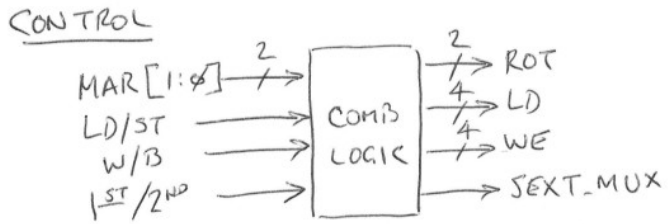
Unaligned Access



LD/ST	MAR[0]	W/B	1ST/2ND	ROT	LD.H	SEXT	LD.L	WE.H	WE.L
LD	0	W	1ST	0	1	0	1	0	0
LD	0	W	1ST	0	1	0	1	0	0
LD	0	B	1ST	1	1	X	1	0	0
LD	1	B	2ND	1	1	X	0	0	0
LD	1	B	1ST	1	1	1	0	0	0
ST	0	W	1ST	0	0	X	0	1	1
ST	0	W	1ST	0	0	X	0	0	1
ST	0	W	2ND	1	0	X	0	1	0
ST	1	W	2ND	0	0	X	0	0	1
ST	1	B	1ST	1	0	X	0	1	0

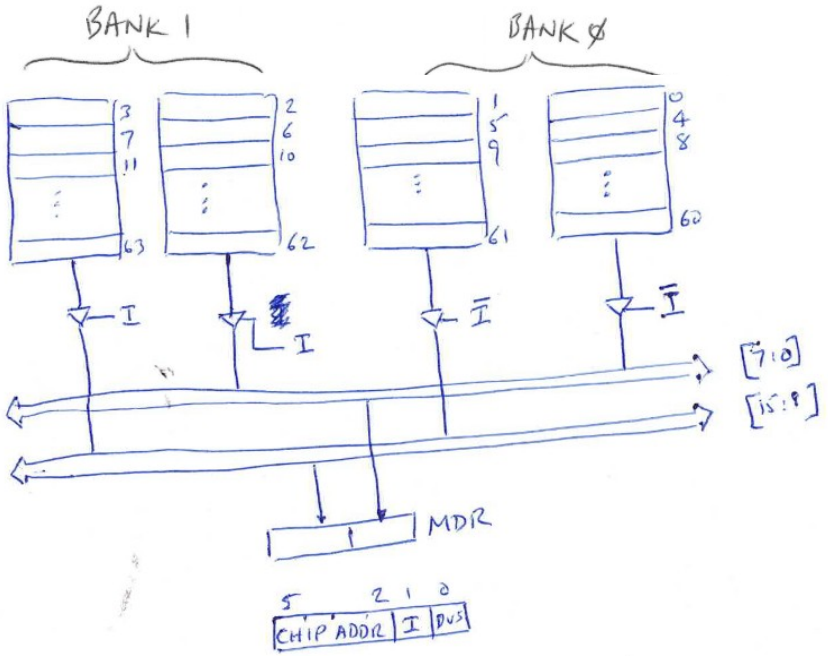
Unaligned Access



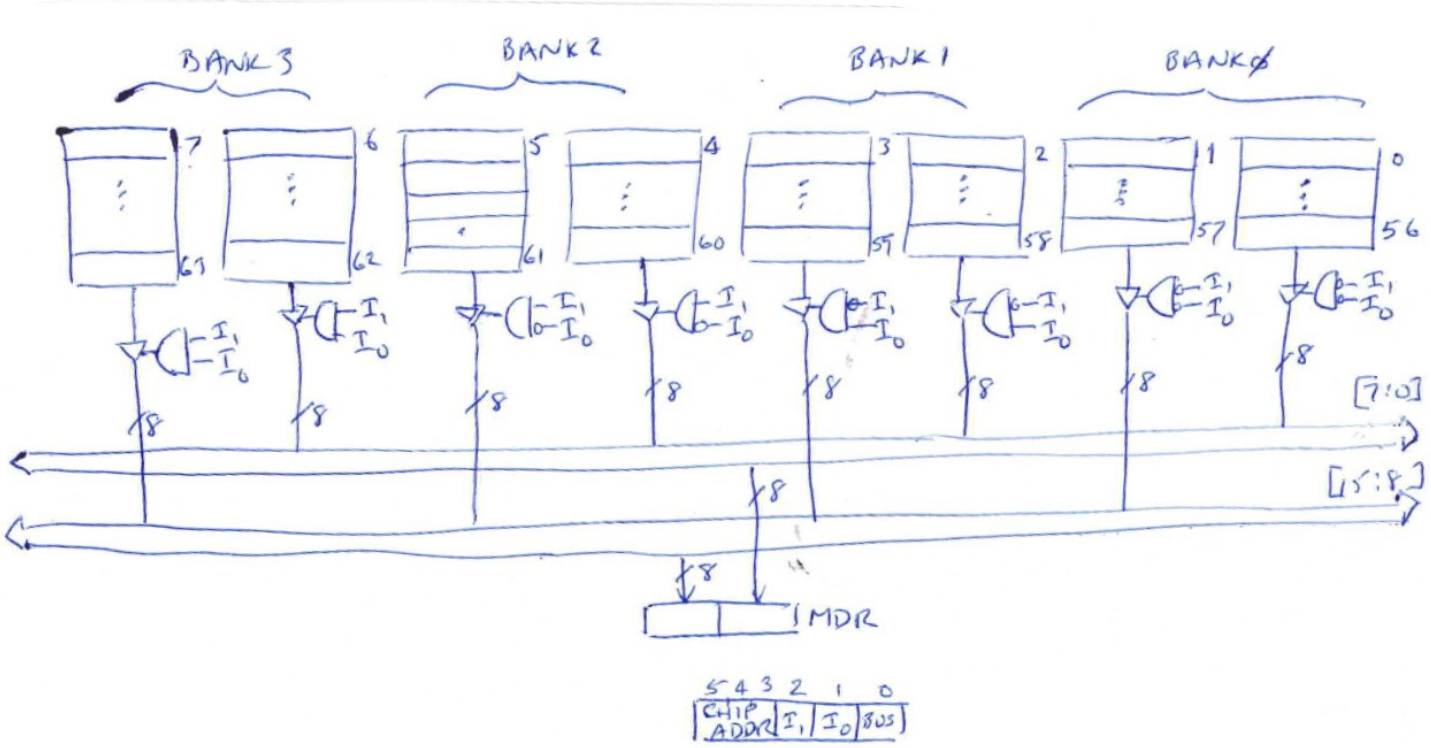


Interleaving

- ***2-way interleaved (i.e., 2 banks)***
- ***64 bytes of memory, using 16 byte chips***
- ***16 bit bus supplied by one of the two banks***



4-way Interleaved

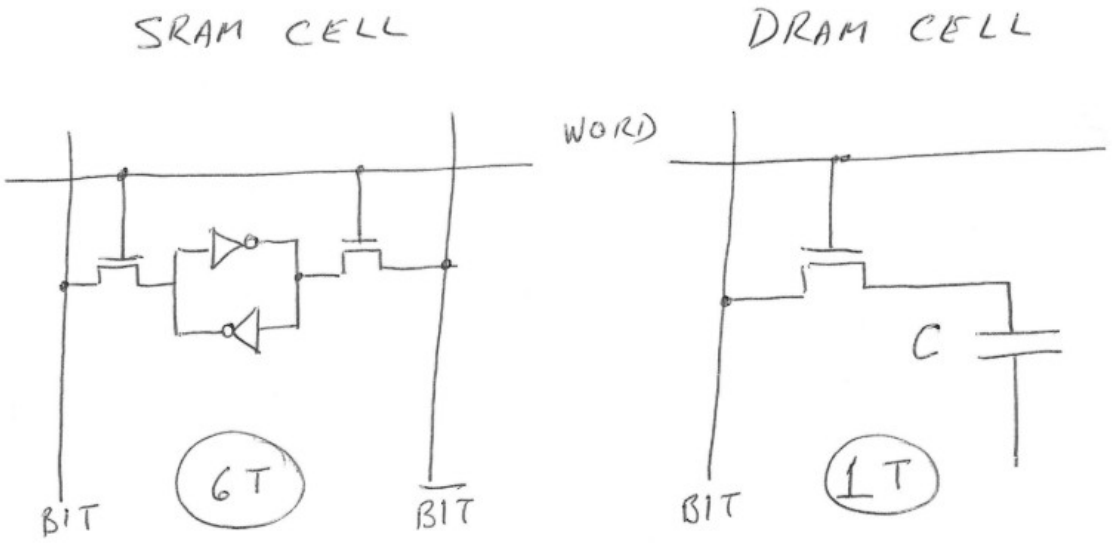


- ***How many cycles to perform the following?***

VLD V1, M[4]

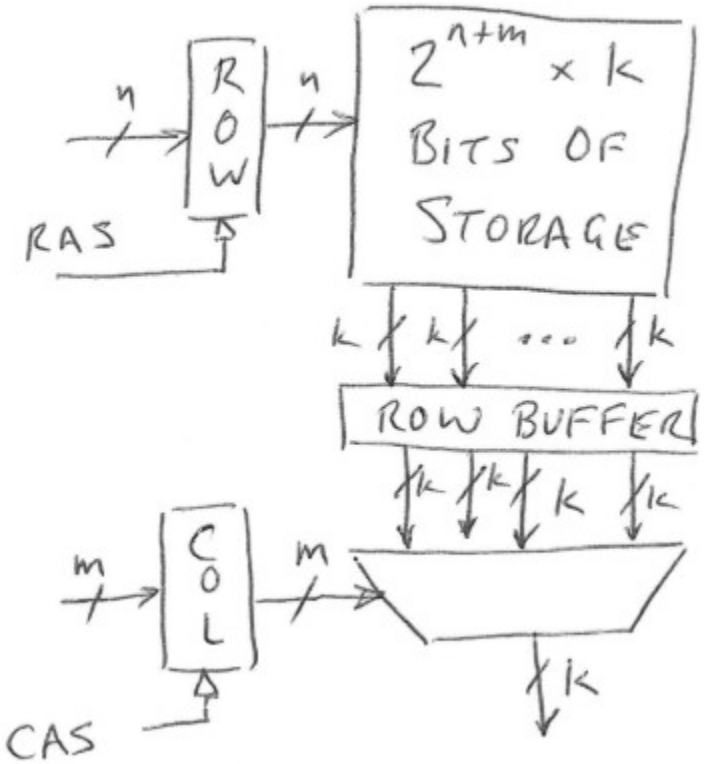
With VL = 6 16-bit words

The Devices and their Tradeoffs

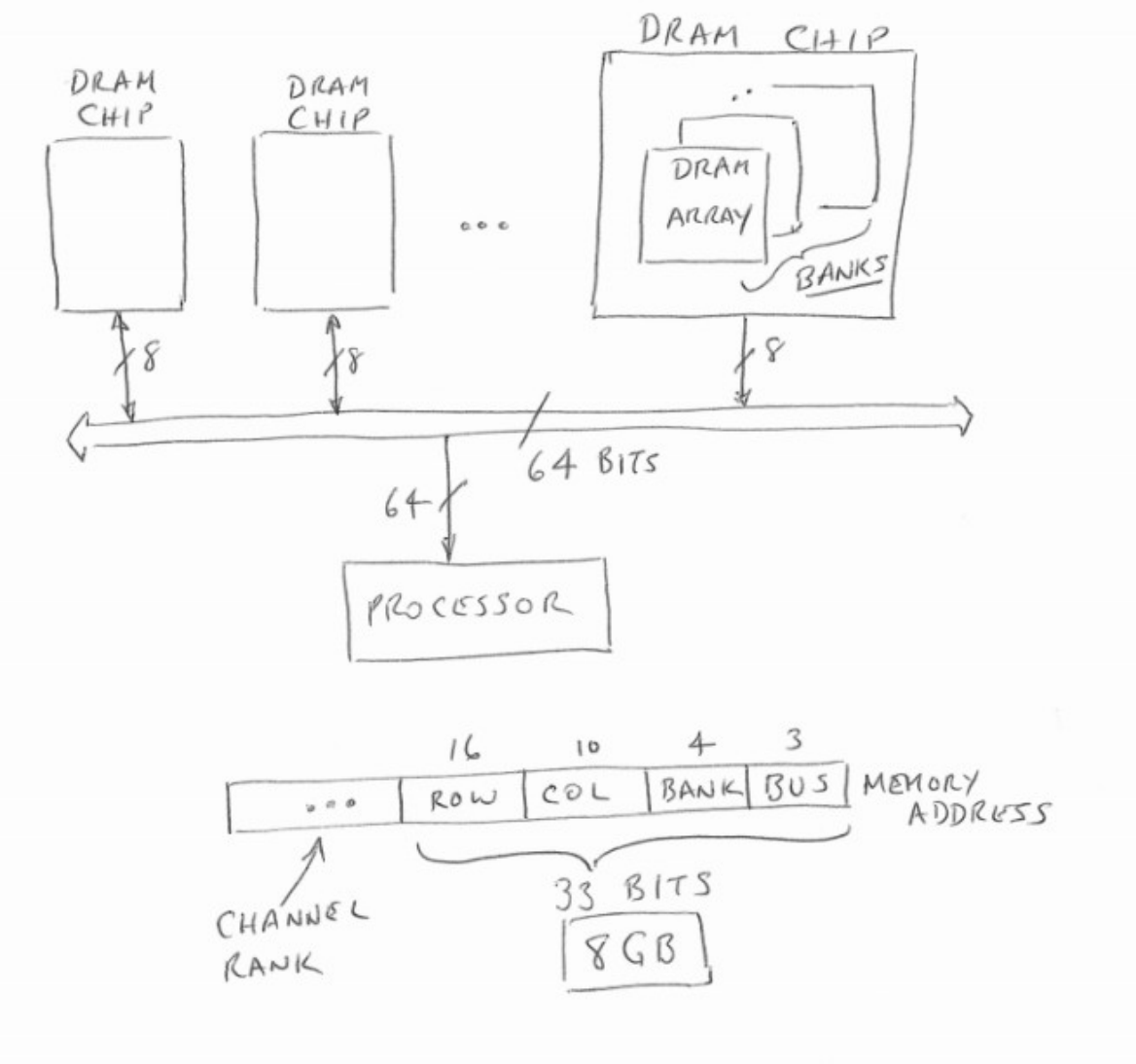


	SRAM	*DRAM*	*NVM*
Latency:	*Low*	*High*	*Highest*
Density:	*Low*	*High*	*Highest*
Persistence:	*Static*	*Dynamic*	*Non-vol*
Refresh:	*No*	*Yes*	*No*

The DRAM Array



DRAM Memory


DRAM CHIP
DRAM CHIP
DRAM CHIP
DRAM ARRAY
BANKS
8
8
8
64 BITS
64
PROCESSOR
16
10
4
3
ROW
COL
BANK
BUS
MEMORY ADDRESS
33 BITS
8 GB
CHANNEL
RANK

The Memory Controller

- ***Determines which access to initiate***
  - ***Bank information***
  - ***Row buffer open/closed, last access R/W***
  - ***Demand vs Prefetch***
- ***One per channel***
- ***Between the core and the DRAMs***

Error Detection/Correction

- ***Parity***
  - ***Detects single bit errors***
  - ***Errors must be statistically independent***
- ***ECC***
  - ***When detecting is not good enough***
  - ***Corrects single bit errors***
  - ***Errors must be statistically independent***
- ***Checksum***
  - ***For large numbers of bits transmitted***
  - ***Errors are not statistically independent***

Parity

- ***Simplest mechanism***
- ***Detects single bit errors if statistically independent***
- ***Typically, for 8 bits of data, we transfer 9 bits***
- ***The 9th bit is the XOR of the 8 information bits***
  - ***Guarantees that the number of 1's transferred is even***
  - ***At destination, count them. If odd, an error has occurred!***
  - ***Retransmit!***

ECC

- ***Errror Correcting Codes (when detecting is not enough***
- ***Allows the correct information to be reconstructed***
- ***We show by an example:***
  - ***We want to transfer n bits (let n=8 in this example))***
  - ***We specify n+logn+1 bits (i.e., 8+3+1 bits) as follows,***
    where Di is a data bit, and Pi is one of the extra logn+1 bits.

Bit:	***12***	***11***	***10***	***09***	***08***	***07***	***06***	***05***	***04***	***03***	***02***	***01***
	D7	***D6***	***D5***	***D4***	***P8***	***D3***	***D2***	***D1***	***P4***	***D0***	***P2***	***P1***

Note the bit number of each bit (e.g., D4 is Bit 9, in binary 1001):

D7	***D6***	***D5***	***D4***	***P8***	***D3***	***D2***	***D1***	***P4***	***D0***	***P2***	***P1***
1	***1***	***1***	***1***	***1***	***0***	***0***	***0***	***0***	***0***	***0***	***0***
1	***0***	***0***	***0***	***0***	***1***	***1***	***1***	***1***	***0***	***0***	***0***
0	***1***	***1***	***0***	***0***	***1***	***1***	***0***	***0***	***1***	***1***	***0***
0	***1***	***0***	***1***	***0***	***1***	***0***	***1***	***0***	***1***	***0***	***1***

ECC (continued)

- ***Continuing…***
  - ***We form four parity (i.e., XOR) functions, one for each row, XORing the bits in each row that has a 1 in its entry.***
    For example, P8 = XOR (D7, D6, D5, D4)
    For example, P4 = XOR (D7, D3, D2, D1)
  - ***At the destination, the four parity functions are examined***
  - ***If any gave an odd number of 1s, it must have been caused by the bit that transmitted in error.***
  - ***We identify that bit by its "bit number," and correct it!***
    e.g., if D4 flipped, it would cause parity errors for P8 and P1, but not P4 or P2. P8(1),P4(0),P2(0),P1(1) identifies 1001, the bit number for D4, so we can correct it.

Checksum

- ***When the probability of error is not statistically independent***
- ***and there is likely to be a burst of bits in error***
- ***Original scheme: use a linear feedback shift register***
  - ***Input bit-serial the information to be transferred***
  - ***Output the bits from the shift register***
  - ***After the input has been output, output the content of LFSR***
  - ***At the destination, repeat the process***
  - ***If an error occurred, it will show up in the LFSR***

Todah!