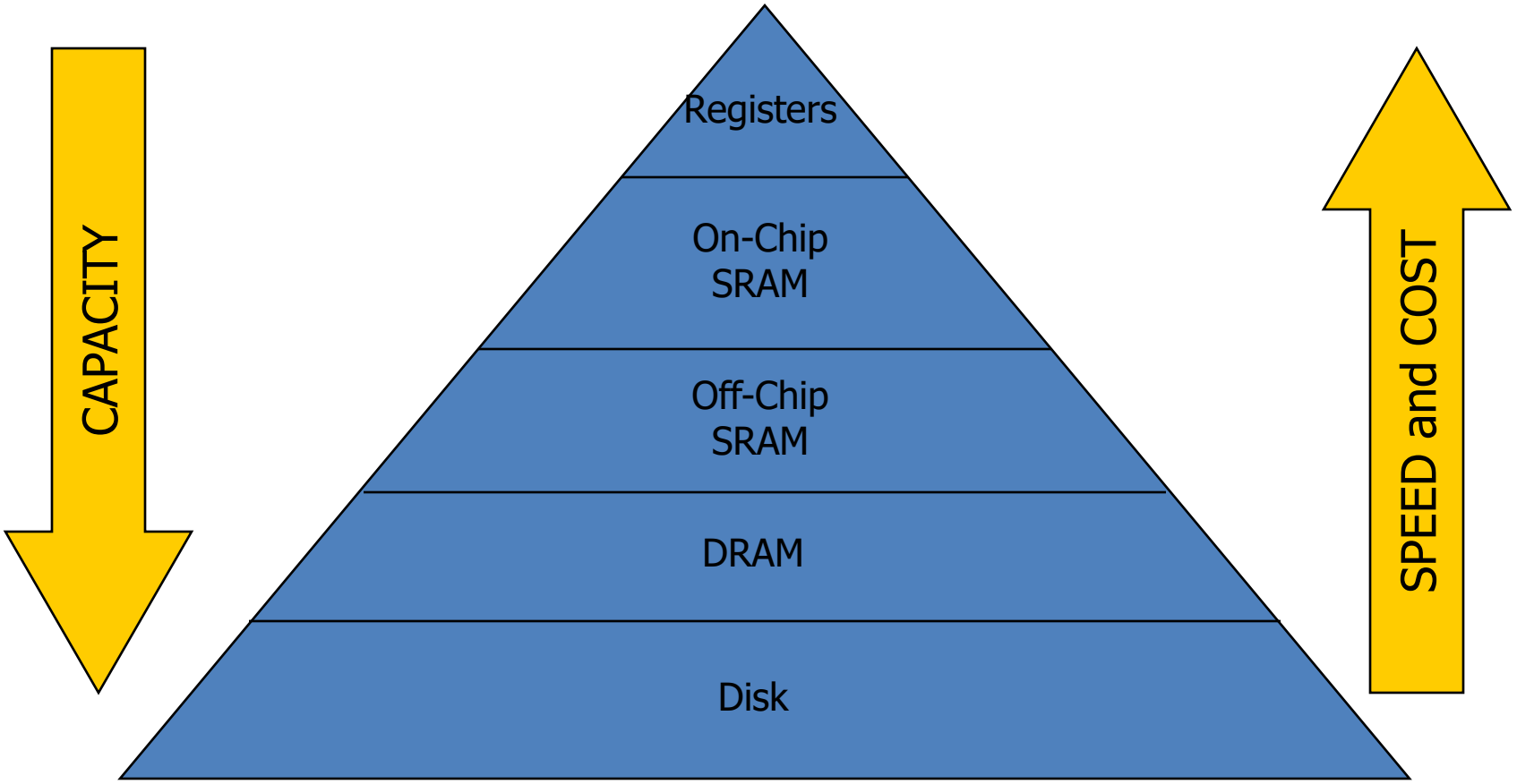
ECE/CS 552:
Main Memory and ECC

© Prof. Mikko Lipasti

Lecture notes based in part on slides created by Mark Hill, David Wood, Guri Sohi, John Shen and Jim Smith

Memory Hierarchy

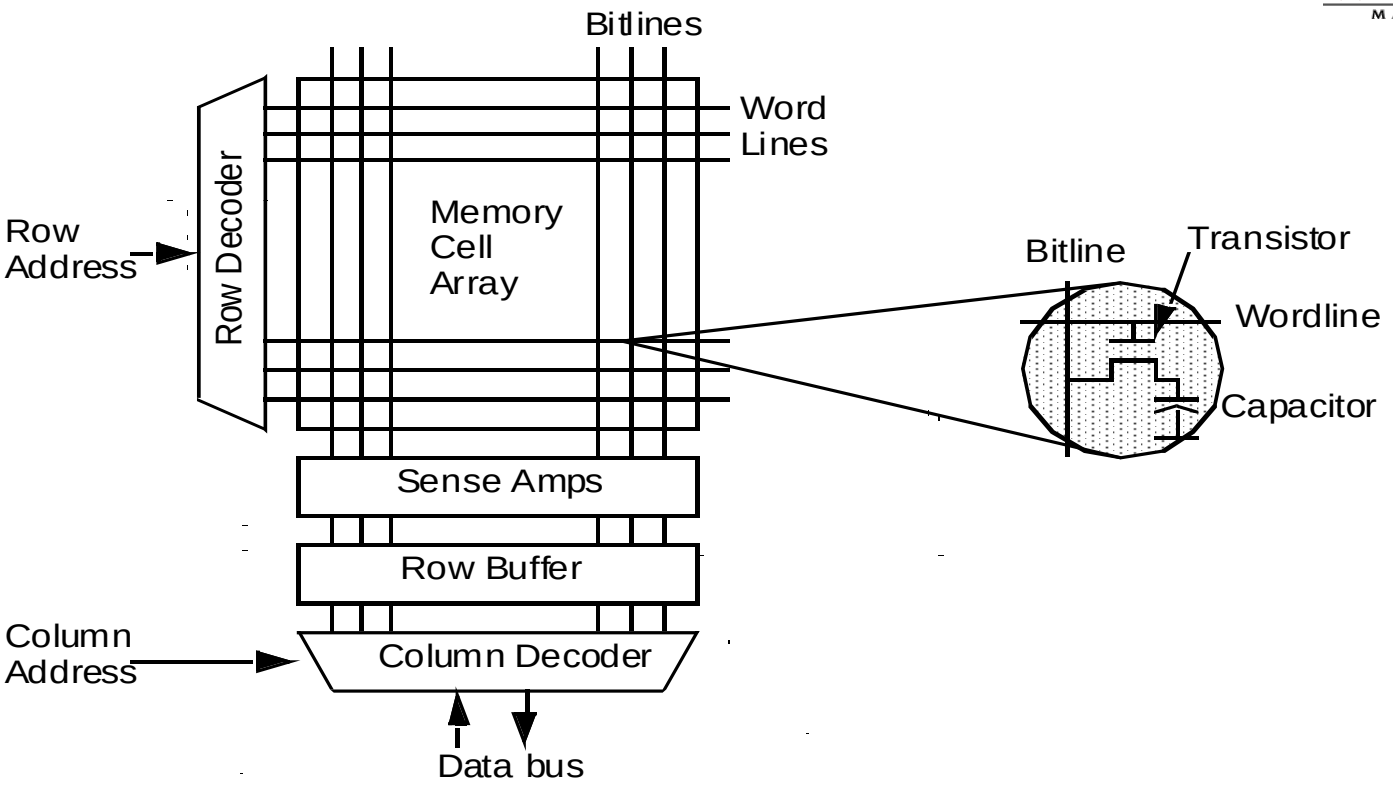


Main Memory Design

THE UNIVERSITY of WISCONSIN MADISON

- Commodity DRAM chips
- Wide design space for
  - Minimizing cost, latency
  - Maximizing bandwidth, storage
- Susceptible to soft errors
  - Protect with ECC (SECDED)
  - ECC also widely used in on-chip memories, busses

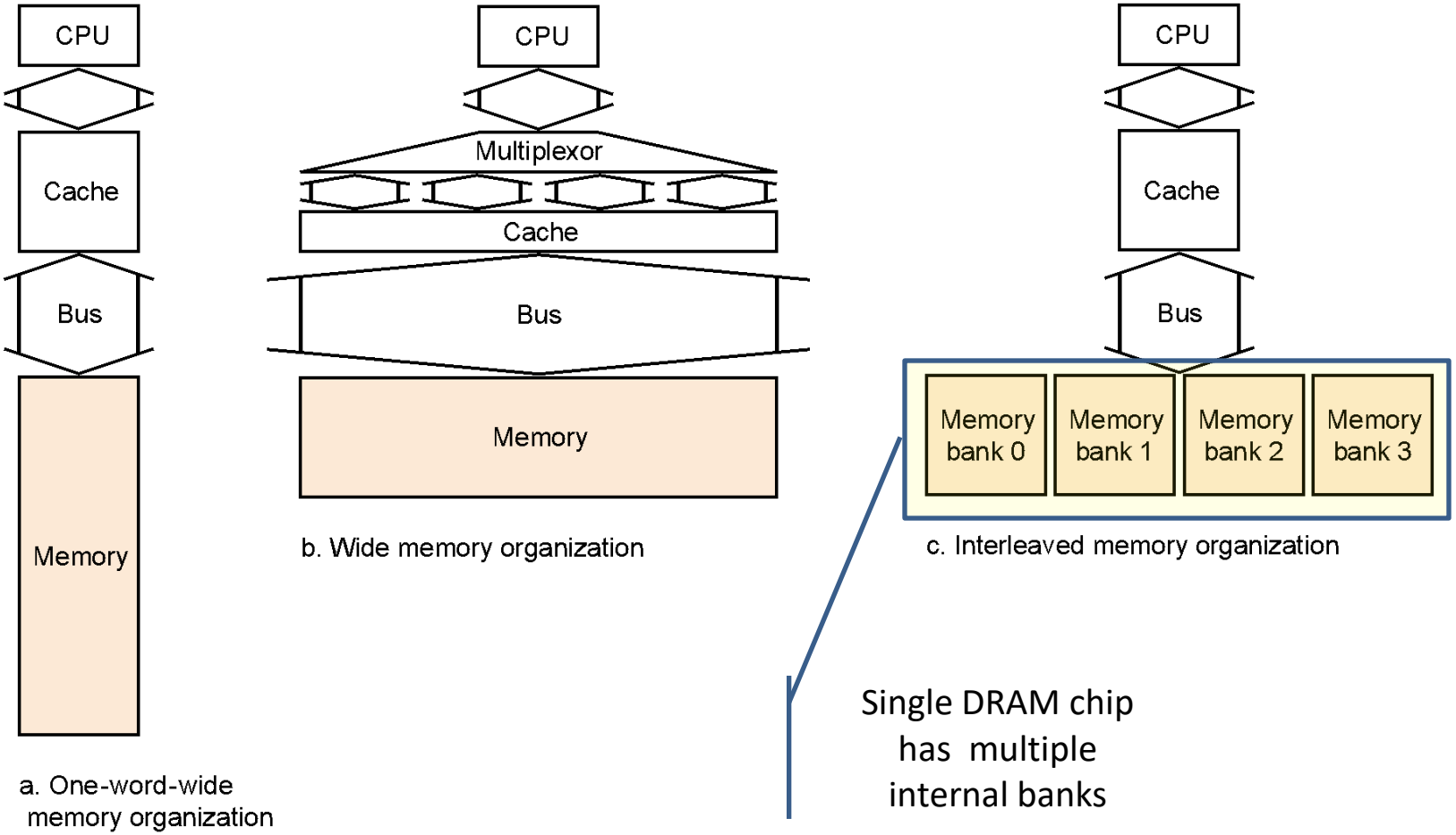
DRAM Chip Organization



- Optimized for density, not speed
- Data stored as charge in capacitor
- Discharge on reads => destructive reads
- Charge leaks over time
  - refresh every 64ms

- Read entire row at once (RAS, page open)
- Read word from row (CAS)
- Burst mode (sequential words)
- Write row back (precharge, page close)

Main Memory Design



Main Memory Access

- Each memory access (DRAM bus clocks, 10x CPU cycle time)
  - 5 cycles to send row address (page open or RAS)
  - 1 cycle to send column address
  - 3 cycle DRAM access latency
  - 1 cycle to send data (CAS latency = 1+3+1 = 5)
  - 5 cycles to send precharge (page close)
  - 4 word cache block
- One word wide: (r=row addr, c= col addr, d=delay, b=bus, p=precharge)

  rrrrrcdddbcdddbcdddbcdddbpppppp

  - 5 + 4 * (1 + 3 + 1) = 25 cycles delay
  - 5 more busy cycles (precharge) till next command

Main Memory Access

- One word wide, burst mode (pipelined)

  rrrrrcdddb

  b

  b

  bpppppp

  – 5 + 1 + 3 + 4 = 13 cycles

  – Interleaving is similar, but words can be from different rows, each open in a different bank

- Four word wide memory:

  rrrrrcdddbpppppp

  – 5 + 1 + 3 + 1 = 9 cycles

Error Detection and Correction

- Main memory stores a huge number of bits
  - Probability of bit flip becomes nontrivial
  - Bit flips (called soft errors) caused by
    - Slight manufacturing defects
    - Gamma rays and alpha particles
    - Electrical interference
    - Etc.
  - Getting worse with smaller feature sizes
- Reliable systems must be protected from soft errors via ECC (error correction codes)
  - Even PCs support ECC these days

Error Correcting Codes

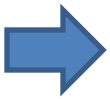
THE UNIVERSITY of WISCONSIN MADISON

- Probabilities:

  P(1 word no errors) > P(single error) > P(two errors) >> P(>2 errors)

- Detection - signal a problem
- Correction - restore data to correct value
- Most common
  - Parity - single error detection
  - SECDED - single error correction; double error detection
- Supplemental reading on course web page!

ECC Codes for One Bit



Power	Correct	#bits	Comments
Nothing	0,1	1	
SED	00,11	2	01,10 detect errors
SEC	000,111	3	001,010,100 => 0 110,101,011 => 1
SECDED	0000,1111	4	One 1 => 0 Two 1's => error Three 1's => 1

ECC

# 1's	0	1	2	3	4
Result	0	0	Err	1	1

- Hamming distance
  - No. of bit flips to convert one valid code to another
  - All legal SECDED codes are at Hamming distance of 4
    - I.e. in single-bit SECDED, all 4 bits flip to go from representation for '0' (0000) to representation for '1' (1111)

ECC

- Reduce overhead by applying codes to a word, not a bit
  - Larger word means higher p(>=2 errors)

# bits	SED overhead	SECDED overhead
1	1 (100%)	3 (300%)
32	1 (3%)	7 (22%)
64	1 (1.6%)	8 (13%)
n	1 (1/n)	1 + $\log_2$ n + a little

64-bit ECC

- 64 bits data with 8 check bits

  dddd…..dccccccccc
- DIMM with 9x8-bit-wide DRAM chips = 72 bits
- Intuition
  - One check bit is parity
  - Other check bits point to
    - Error in data, or
    - Error in all check bits, or
    - No error

ECC

- To store (write)
  - Use $data_0$ to compute $check_0$
  - Store $data_0$ and $check_0$
- To load
  - Read $data_1$ and $check_1$
  - Use $data_1$ to compute $check_2$
  - Syndrome = $check_1$ xor $check_2$
    - I.e. make sure check bits are equal

ECC Syndrome

THE UNIVERSITY of WISCONSIN MADISON

Syndrome	Parity	Implications
0	OK	$data_1$==$data_0$
n != 0	Not OK	Flip bit n of $data_1$ to get $data_0$
n != 0	OK	Signals uncorrectable error

4-bit SECDED Code

Bit Position	001	010	011	100	101	110	111	
Codeword	C_1	C_2	b_1	C_3	b_2	b_3	b_4	P
C_1	X		X		X		X	
C_2		X	X			X	X	
C_3				X	X	X	X	
P	X	X	X	X	X	X	X	X

$$C_1 = b_1 \oplus b_2 \oplus b_4$$
$$C_2 = b_1 \oplus b_3 \oplus b_4$$
$$C_3 = b_2 \oplus b_3 \oplus b_4$$
$$P = even\_parity$$

- C_n parity bits chosen specifically to:
  - Identify errors in bits where bit n of the index is 1
  - C_1 checks all odd bit positions (where LSB=1)
  - C_2 checks all positions where middle bit=1
  - C_3 checks all positions where MSB=1
- Hence, nonzero syndrome points to faulty bit

4-bit SECDED Example

$$C_1 = b_1 \oplus b_2 \oplus b_4$$
$$C_2 = b_1 \oplus b_3 \oplus b_4$$
$$C_3 = b_2 \oplus b_3 \oplus b_4$$
$$P = even\_parity$$

Bit Position	1	2	3	4	5	6	7		
Codeword	C_1	C_2	b_1	C_3	b_2	b_3	b_4	P	
Original data	1	0	1	1	0	1	0	0	Syndrome
No corruption	1	0	1	1	0	1	0	0	0 0 0, P ok
1 bit corrupted	1	0	0	1	0	1	0	0	0 1 1, P !ok
2 bits corrupted	1	0	0	1	1	1	0	0	1 1 0, P ok

- 4 data bits, 3 check bits, 1 parity bit
- Syndrome is xor of check bits $C_{1\text{-}3}$
  - If (syndrome==0) and (parity OK) => no error
  - If (syndrome != 0) and (parity !OK) => flip bit position pointed to by syndrome
  - If (syndrome != 0) and (parity OK) => double-bit error

Summary

- Commodity DRAM chips
- Wide design space for
  - Minimizing cost, latency
  - Maximizing bandwidth, storage
- Susceptible to soft errors
  - Protect with ECC (SECDED)
  - ECC also widely used in on-chip memories, busses