# Lec 5: Memory and a Simple Processor

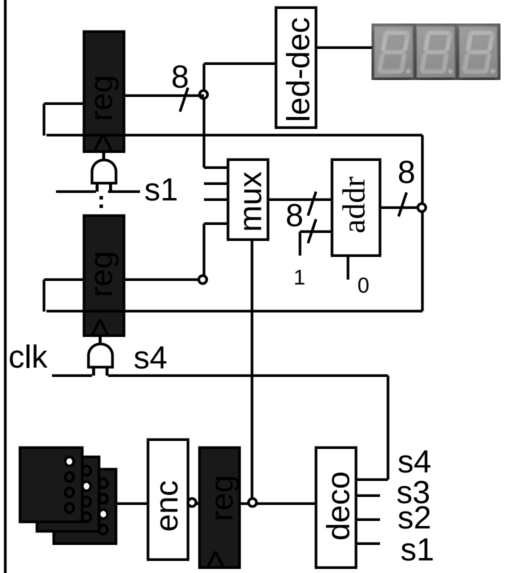
**Kavita Bala**

**CS 3410, Fall 2008**

Computer Science

Cornell University

## A Vote Counter



- Encode
- Decode
- Register file of counts
- Data values flow from register file
- To addition unit
- Back to register file

© Kavita Bala, Computer Science, Cornell University

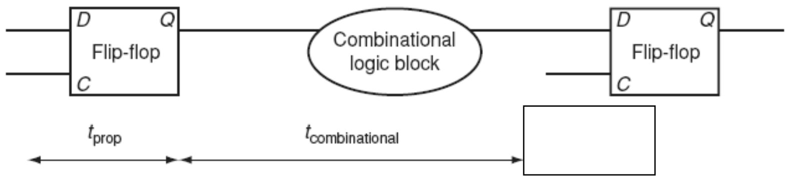
## Two Questions

- Addition is very important
  - Is this design fast enough?
  - Also, is this design general enough?

© Kavita Bala, Computer Science, Cornell University

## Performance

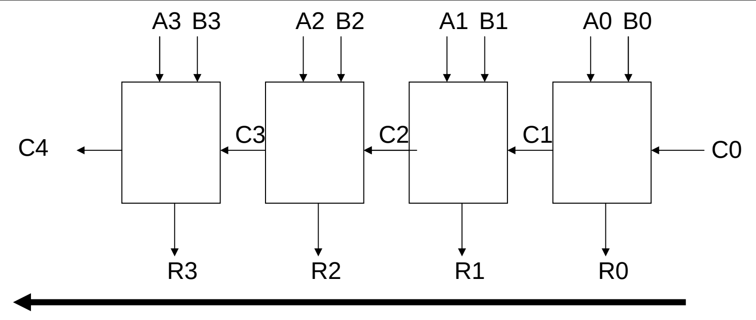
- Speed of a circuit is affected by the number of gates in series (on the *critical path* or the *deepest level of logic*)



- The speed of a gate is affected by the number of inputs

© Kavita Bala, Computer Science, Cornell University

## 4-bit Ripple Carry Adder



Carry ripples from right to left

- First adder, 2 gate delay
- Second adder, 2 gate delay

© Kavita Bala, Computer Science, Cornell University

## Observations

- Have to wait for Cin
- Can we compute in parallel in some way?
- CLA carry look-ahead adder

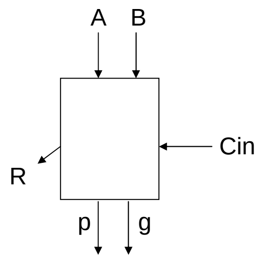
© Kavita Bala, Computer Science, Cornell University

## Carry Look Ahead Logic

- Can we reason independent of Cin?
  - Just based on (A,B) only
- When is Cout == 1, irrespective of Cin
- If Cin == 1, when is Cout also == 1

© Kavita Bala, Computer Science, Cornell University

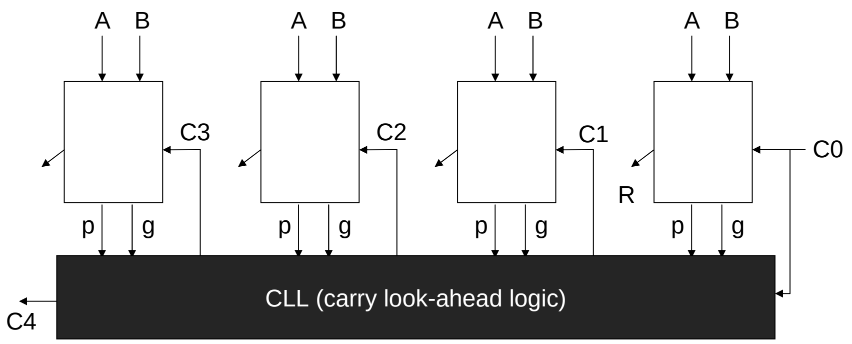
## 1-bit CLA adder



- Create two terms: propagator, generator
- g = 1, generates Cout: g = AB
  - Irrespective of Cin
- p = 1, propagates Cin to Cout: p = A xor B
- p and g generated in 1 cycle delay
- R is 2 cycle delay after we get Cin

© Kavita Bala, Computer Science, Cornell University

## 4-bit CLA



- CLL takes p,g from all 4 bits, C0 and generates all Cs
  - 2 gate delay

C1=g0+p0C0

C2=g1+p1C1

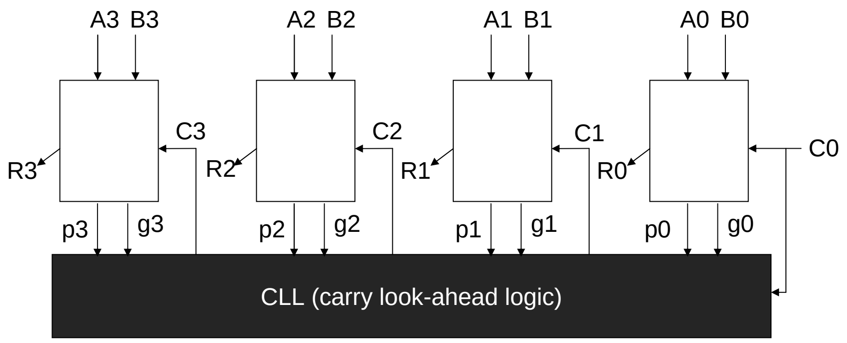
= g1 + p1(g0+p0C0) = g1+p1g0+p1p0C0

C3=g2+p2g1+p2p1g0+p2p1p0c0

C4=g3+p3g2+p3p2g1+p3p2p1g0+p3p2p1p0c0

© Kavita Bala, Computer Science, Cornell University

## 4-bit CLA



- Given A,B's, all p,g's are generated in 1 gate delay in parallel.
- Given all p,g's, all C's are generated in 2 gate delay in parallel.
- Given all C's, all R's are generated in 2 gate delay in parallel.

- Sequential operation in RCA is made into parallel operation!!

© Kavita Bala, Computer Science, Cornell University

## Ripple Carry vs Carry Lookahead

- Ripple carry adder vs Carry lookahead adder for 8 or 16 bits
  - 2 x 8 vs. 5
  - 2 x 16 vs. 5

- Can’t do it for all 32 bits though
- So use hierarchical construction

© Kavita Bala, Computer Science, Cornell University

## Memory

## Register File

- Set of registers
  - – Read or written
  - – Use register number to access it
- Read or write ports
  - – Decoder for each port
- D or SR flip flops to store bits

© Kavita Bala, Computer Science, Cornell University

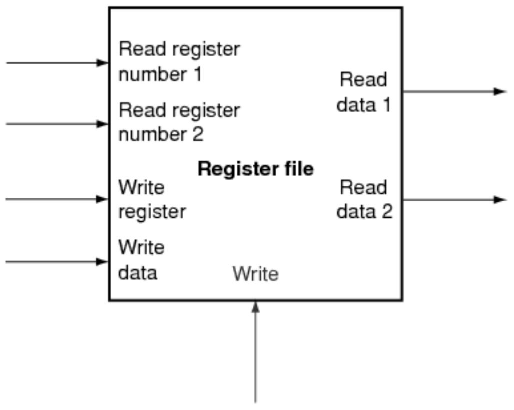
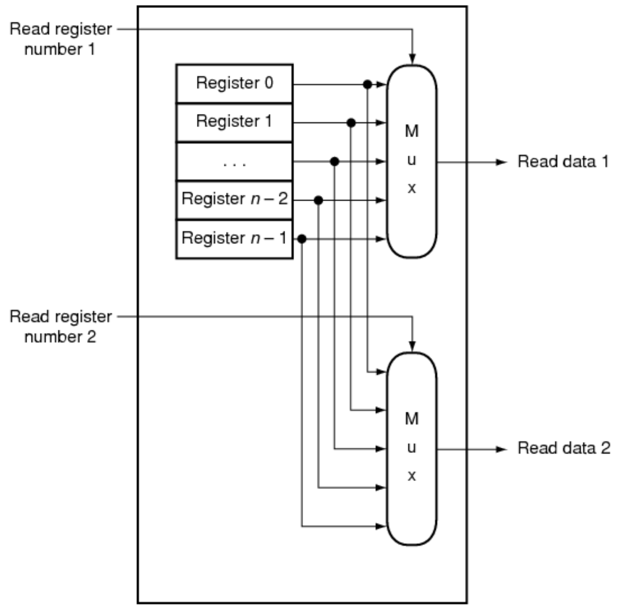
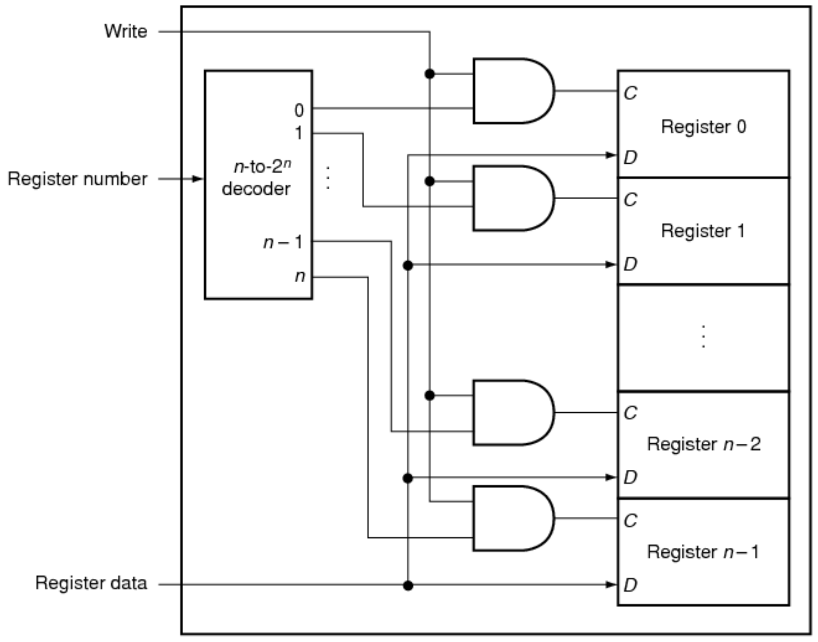


**FIGURE B.8.7 A register file with two read ports and one write port has five inputs and two outputs.** The control input Write is shown in color.

© Kavita Bala, Computer Science, Cornell University



© Kavita Bala, Computer Science, Cornell University



© Kavita Bala, Computer Science, Cornell University

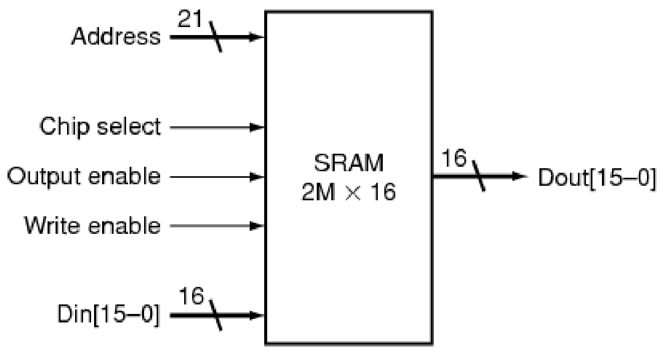
## Memory

- Various technologies
  - S-RAM, D-RAM, NV-RAM
- Non-Volatile RAM
  - Data remains valid even through power outages
  - More expensive
  - Limited lifetime; after 100000 to 1M writes, NV-RAM degrades
  - Flash cards

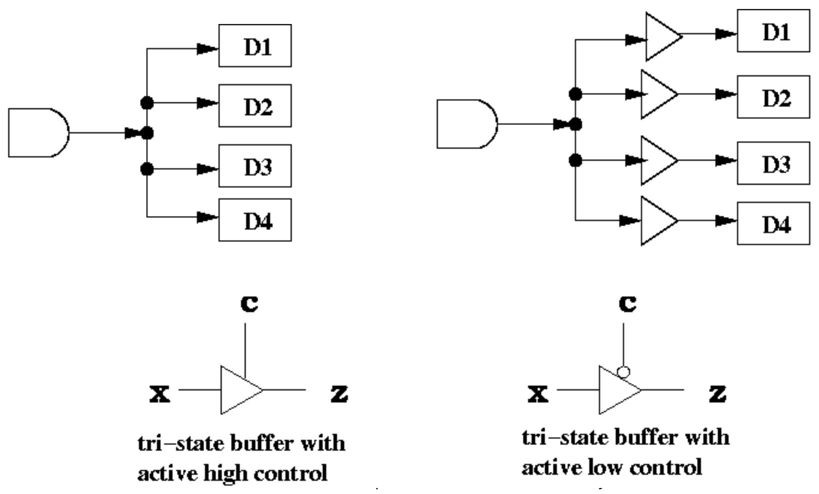
© Kavita Bala, Computer Science, Cornell University

## Static RAM: SRAM

- Static-RAM
  - So called because once stored, data values are stable as long as electricity is supplied
  - Based on regular flip-flops with gates



## Tristate Buffers


D1
D2
D3
D4
c
x
z
tri–state buffer with active high control
tri–state buffer with active low control

## Tristate Buffers

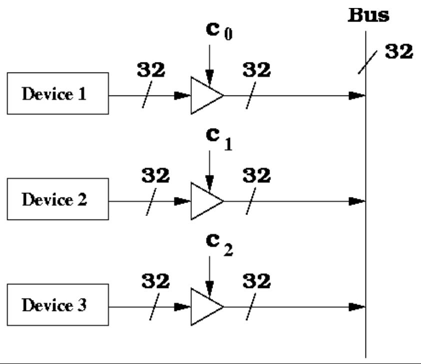
- 3 states
  - – asserted (1)
  - – deasserted (0)
  - – high impedance (Z)

| c | x | z |
|---|---|---|
| 0 | 0 | Z |
| 0 | 1 | Z |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

© Kavita Bala, Computer Science, Cornell University

## How to build large memories?

- Cannot use a 2M->1 multiplexer!
- Use a shared line (called bit line)
- Can create a bus out of multiple bit lines
- Only one of the inputs can drive the output



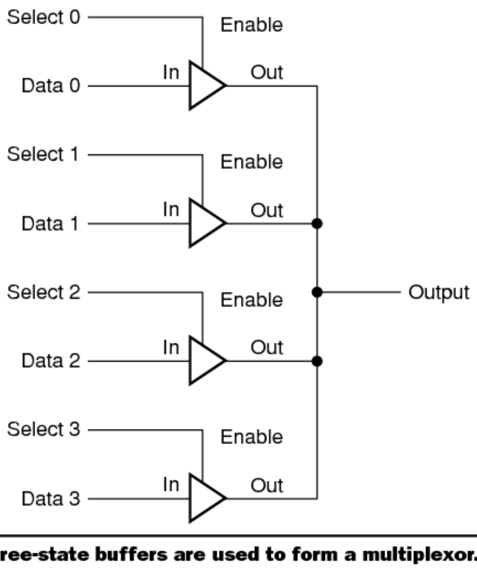


**FIGURE B.9.2 Four three-state buffers are used to form a multiplexor.** Only one of the four Select inputs can be asserted. A three-state buffer with a deasserted Output enable has a high-impedance output that allows a three-state buffer whose Output enable is asserted to drive the shared output line.

© Kavita Bala, Computer Science, Cornell University

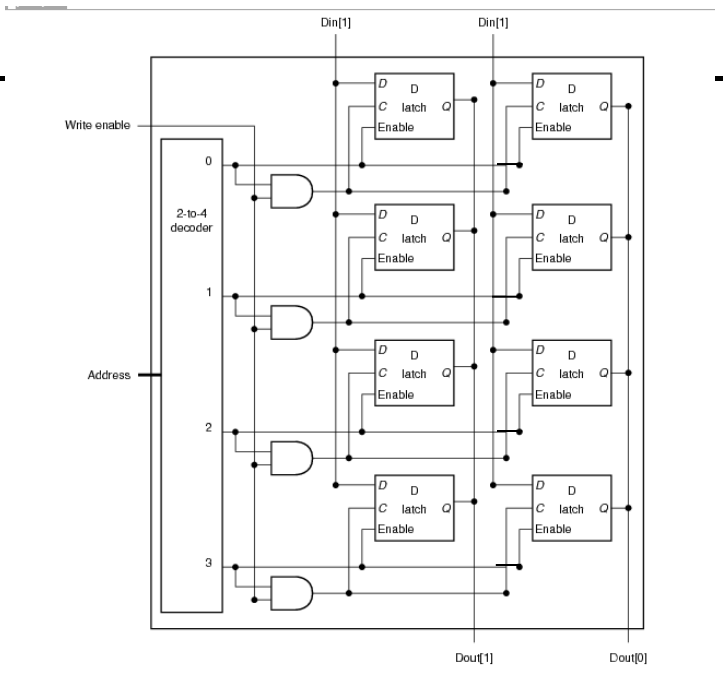


**FIGURE B.9.3 The basic structure of a 4 × 2 SRAM consists of a decoder that selects which pair of cells to activate.** The acti-

## Big Memories

- Tri state buffer got rid of big mux
- But still need a big decoder to pick the right entry
  - 2M x 16 SRAM requires
    - 21 to 2M decoder
    - And 2M lines!
- Instead
  - Rectangular arrays
  - 2-step decode

© Kavita Bala, Computer Science, Cornell University

## Parallel Memory Banks

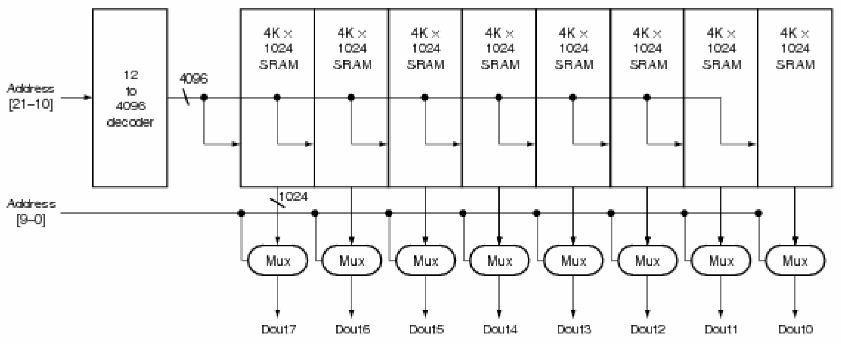


**FIGURE B.9.4 Typical organization of a 4M × 8 SRAM as an array of 4K × 1024 arrays.** The first decoder generates the addresses for eight 4K × 1024 arrays; then a set of multiplexors is used to select 1 bit from each 1024-bit-wide array. This is a much easier design than a single-level decode that would need either an enormous decoder or a gigantic multiplexor. In practice, a modern SRAM of this size would probably use an even larger number of blocks, each somewhat smaller.

© Kavita Bala, Computer Science, Cornell University

## SRAM

- Needs a few gates per cell
- Used for caches (we talk about this later)
- For higher density, use DRAM

© Kavita Bala, Computer Science, Cornell University

## Dynamic RAM: DRAM

- Dynamic-RAM
  - Data values require constant refresh
  - Internal circuitry keeps capacitor charges

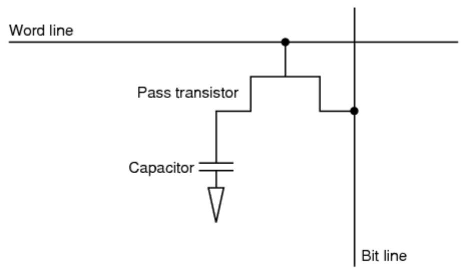


**FIGURE B.9.5 A single-transistor DRAM cell contains a capacitor that stores the cell contents and a transistor used to access the cell.**

© Kavita Bala, Computer Science, Cornell University

## DRAM

- Single transistor vs. many gates
  - Denser and cheaper
- But, need refresh
  - Read and write back
  - Every few milliseconds…
  - Also organized in 2D grid, so can do rows at a time
  - Done independently on chip
- Hence, slower

© Kavita Bala, Computer Science, Cornell University

## Summary

- We now have enough building blocks to build machines that can perform non-trivial computational tasks

- SRAM: caches
- DRAM: main memory

© Kavita Bala, Computer Science, Cornell University