EECS 427
Lecture 5: Speedpath Schematics, Register Files (CAD3), Project Architecture

Last Time

- Euler path layout method
  - To create continuous strips of active
  - Semi-automated
- Registers/Timing
  - Setup, hold time constraints in big picture
  - Setup time measurement technique
  - Some variations on Register styles.

This time

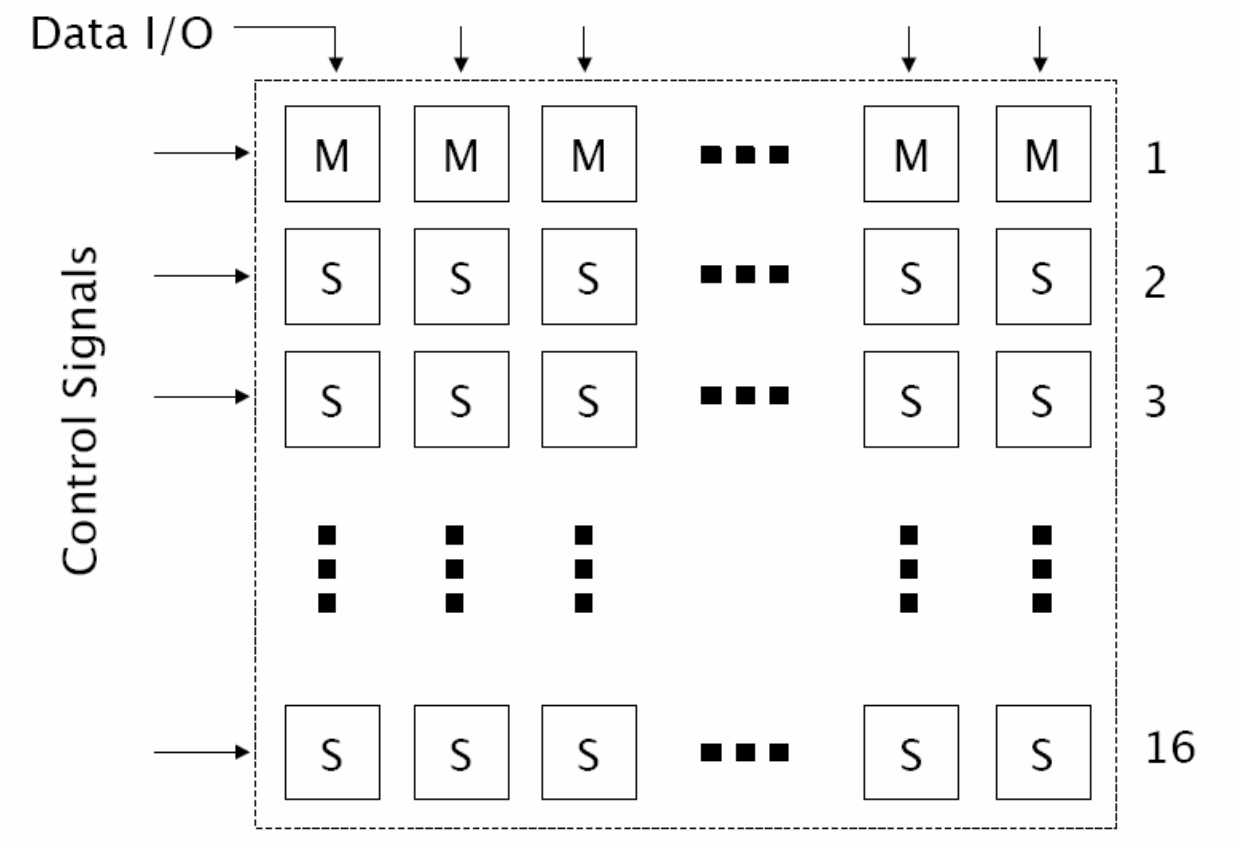
- CAD3 / Register Files
- Speedpath schematics
- Project Architecture

CAD3

- Group project
- Register file
- 16 entry x 16 bits wide
- Must include control signal drivers
- Drivers and cell output drivers perfect candidates for sizing (high load cap)

Four basic RF designs

- Each cell is edge triggered register
- One master with 16 slaves (x 16 wide)
- 16 masters with 2 slaves (x 16 wide)
- SRAM array (more complicated but potentially smallest)

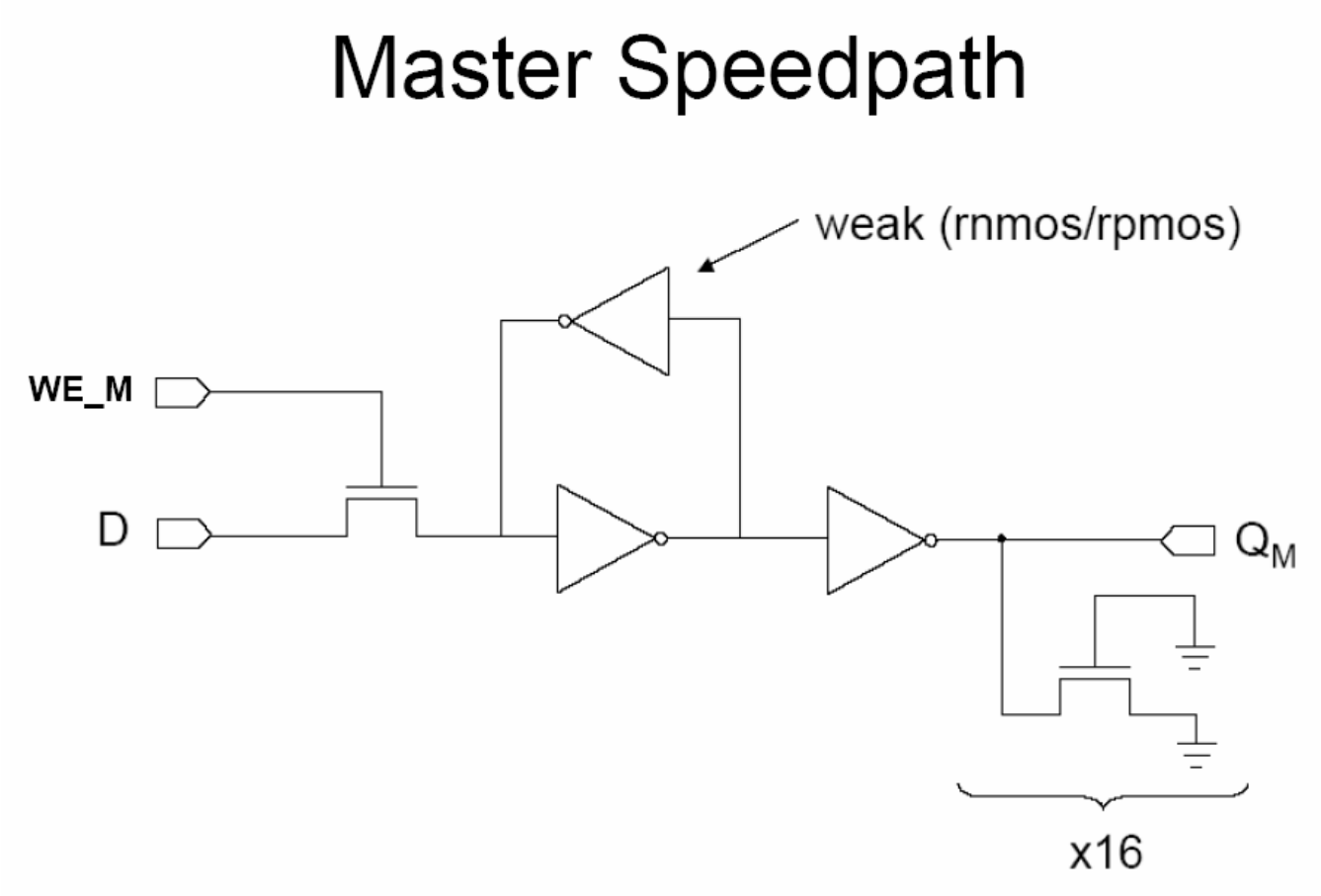

Data I/O
Control Signals
M
S
1
2
3
16

Speedpath Schematic

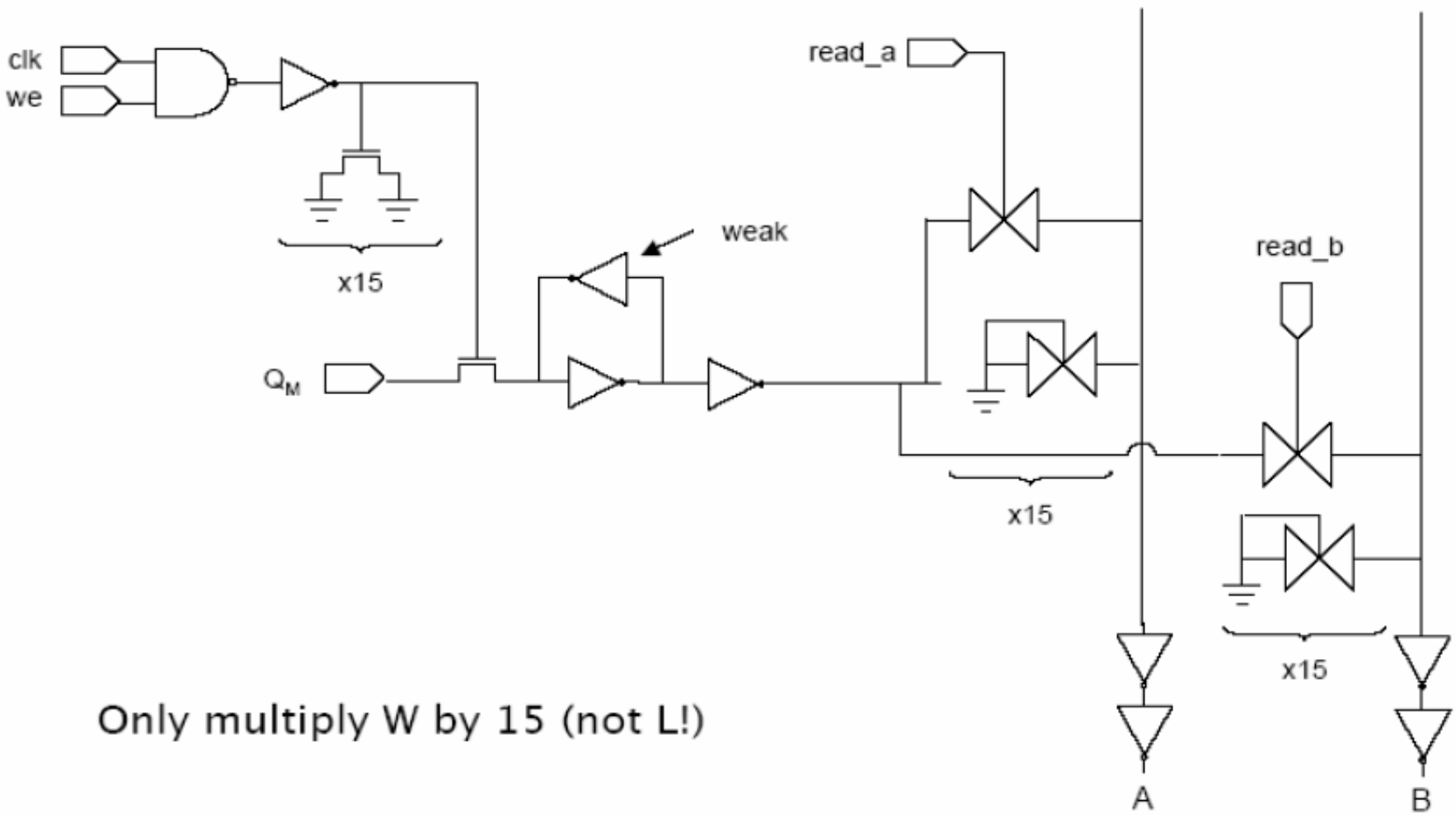
- Used for sizing
- Important for CAD3
- Only useful for regular circuits

- Basic idea : isolate one path through a complex structure, modeling the rest of the structure and size that path.

Speedpath example



Slave Speedpath



Only multiply W by 15 (not L!)

Speedpath schematic

- Don’t forget to add routing cap
- Don’t forget to size related nodes with size on path (you are sizing them all)
- Remember that you are using fake transistors to model cap, don’t be fooled into thinking of them being active
- Feel free to put both master and slave into 1 speedpath schematic

HW 2

- On Tuesday turn in:
- A list of 4 group members :
  - Names, uniqnames, backgrounds
- A name for your group

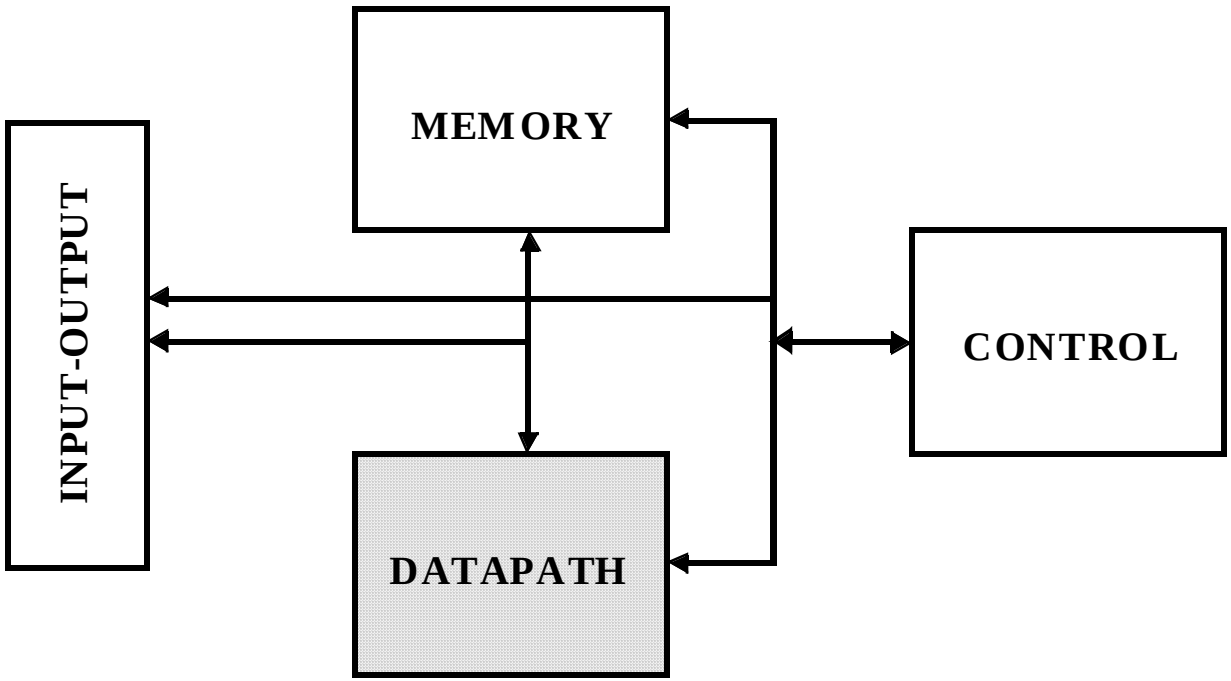
Project architecture

- 2-stage pipeline, 1 word per instruction
  - 1st stage of pipe: instruction fetch (IF)
  - 2nd stage: instruction decode (ID), execute (EX)
  - You can alter this but it's not as easy as it looks
- 16-bit words, with four 4-bit components
  - Most significant 4 bits are the operation code (opcode)
  - Tells which instruction (e.g., ADD, MOV, STOR) is to be performed
  - Next 4 bits give the register address to which the result of the instruction should be written (with a few exceptions)
  - Next 8 bits can contain several pieces of information:
    - Immediate data to be acted upon (rather than accessing this data from a register location)
    - Opcode extensions (since there are more than 2^4 or 16 ops)
    - Address of source register to draw data from

Example instructions

- Direct vs. immediate instructions
- *Add* Rsrc Rdest
  - Rdest ← Rdest *Add* Rsrc
  - Where Rdest and Rsrc are register addresses
- *Add* Imm Rdest
  - Rdest ← Rdest *Add* Imm
  - Where Imm is 8 bits of data (not an address)
- Typical instructions:
  - MOV moves data from 1 reg location to another
  - LOAD loads data from memory to the RF
  - STOR writes data to memory
  - Control flow instructions (conditional branches, jumps, jump and link)
- Look over baseline instructions and extra instructions, think about target application
- Weste 2nd edition (Ch. 9.2, handout coming) is useful as overview of a processor architecture (note it does not exactly reflect our own architecture)

A Generic Digital Processor



Building Blocks for Digital Architectures

Arithmetic unit

- Bit-sliced datapath (adder, multiplier, shifter, comparator, etc.)

Memory

- RAM, ROM, Buffers, Shift registers

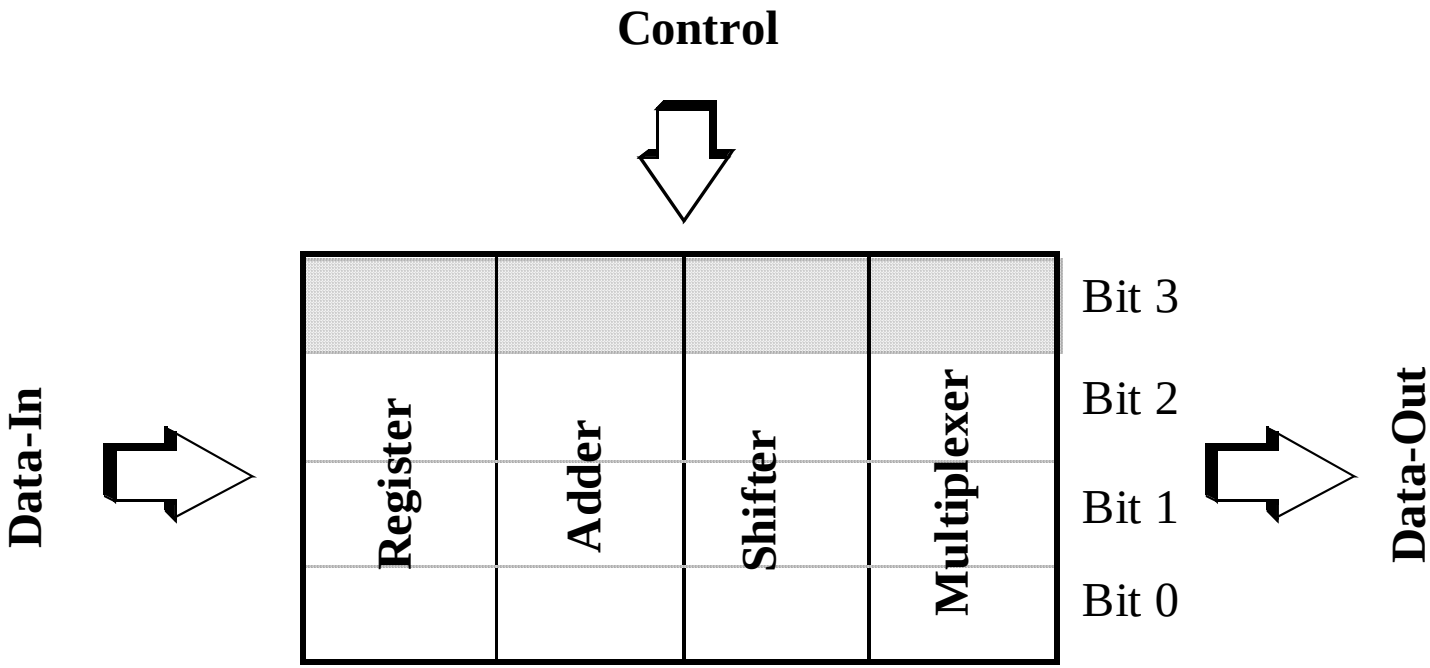
Control

- Finite state machine (PLA, random logic)
- Counters

Interconnect

- Switches
- Arbiters
- Bus

Bit-Sliced Design



Tile identical processing elements

Next Time

- Logical Effort (a formal and relatively simple method of sizing)