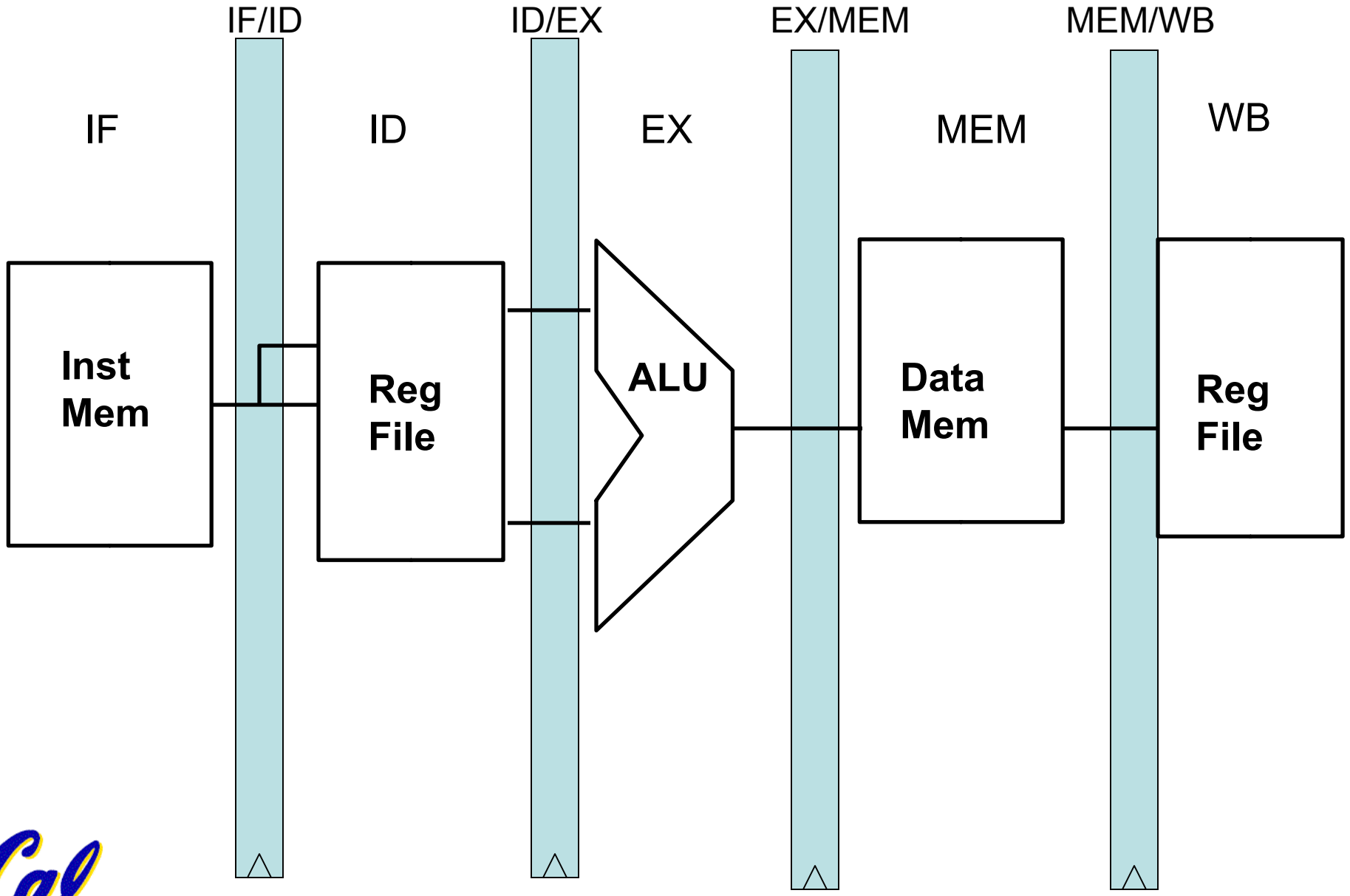
Lab Lecture for Lab #4
Pipelining Your Processor

Jack Kang

cs152-ta@imail.eecs.berkeley.edu

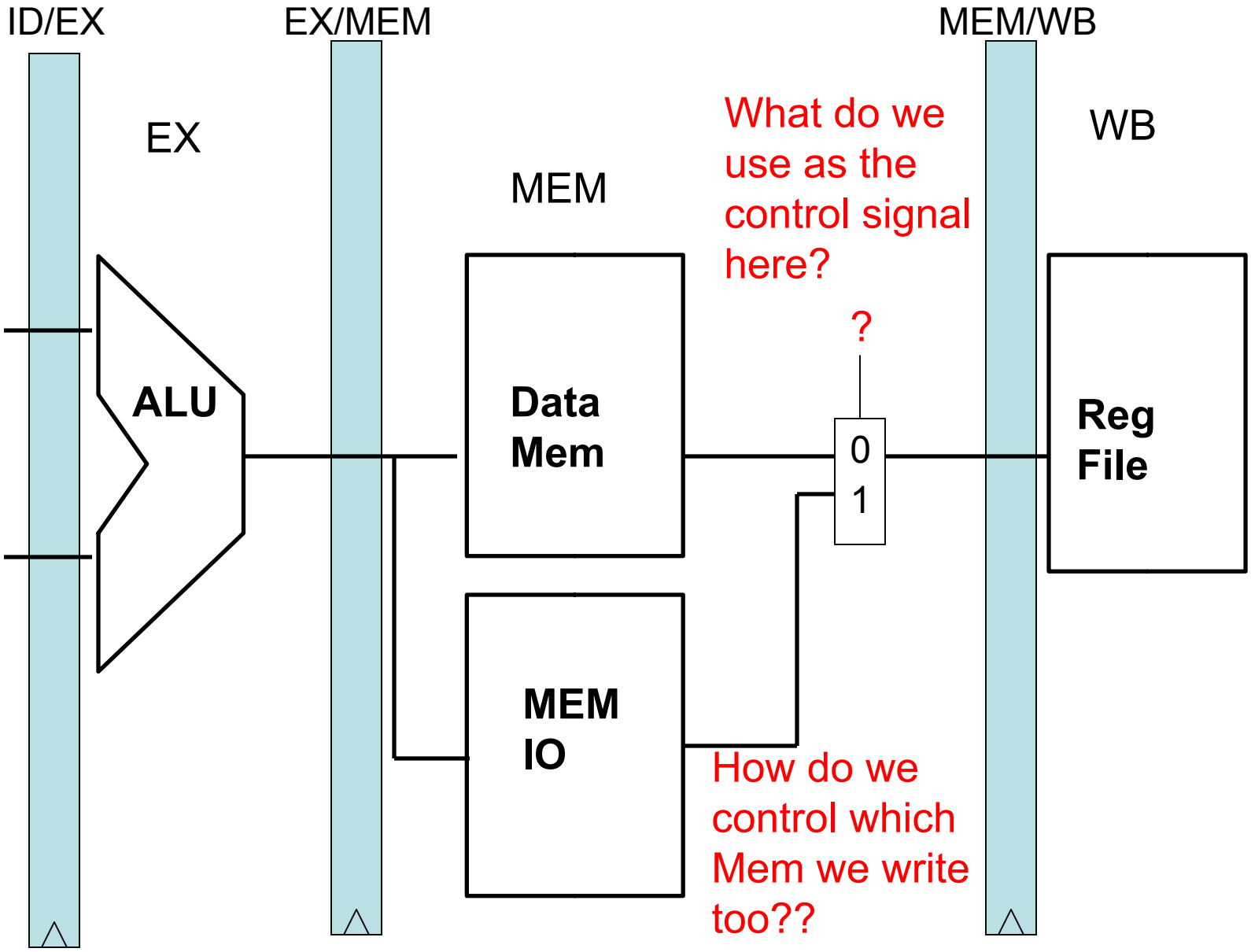
A Pipelined Processor



Memory-Mapped IO

- Any store or writes to memory when the top address bit is 1 will write to your Memory-Mapped IO rather than normal memory.
  - Allows us to display information to HEX LEDs using SW
  - Allows us to take input from the board using LW
  - During simulation, the I/O space is simulated by using text files.

Memory-Mapped IO



Memory-Mapped IO

- Internally, there are only 2 registers.
- The only two valid addresses you can write to are 0xFFFFFFF0 and 0xFFFFFFF4.
  - Note that 0xFFFFFFF4 is -12….
- Other writes to memio can be ignored.
- You may want to make use of the Verilog syntax:
  - `ifdef synthesis, `else, `endif

Adding the Multiplier

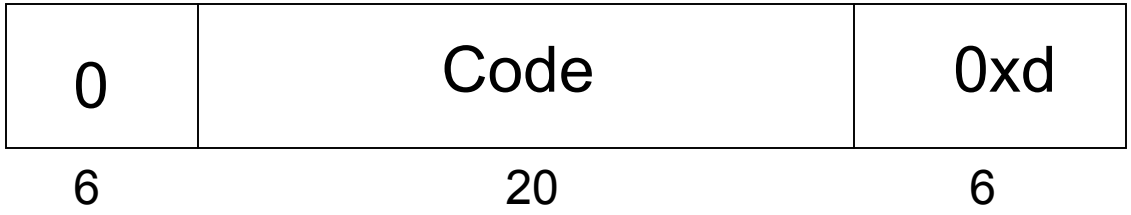
- The multiplier is a “coprocessor”
  - It executes on its own, and the processor doesn’t stall unless it gets another multu, mfhi, or mflo.
  - What signal do we need to add to allow the processor to stall?
  - What stage do we send the signals to the multiplier?

The Break Instruction

add $1, $2, $3

break 5

Add $4, $5, $6

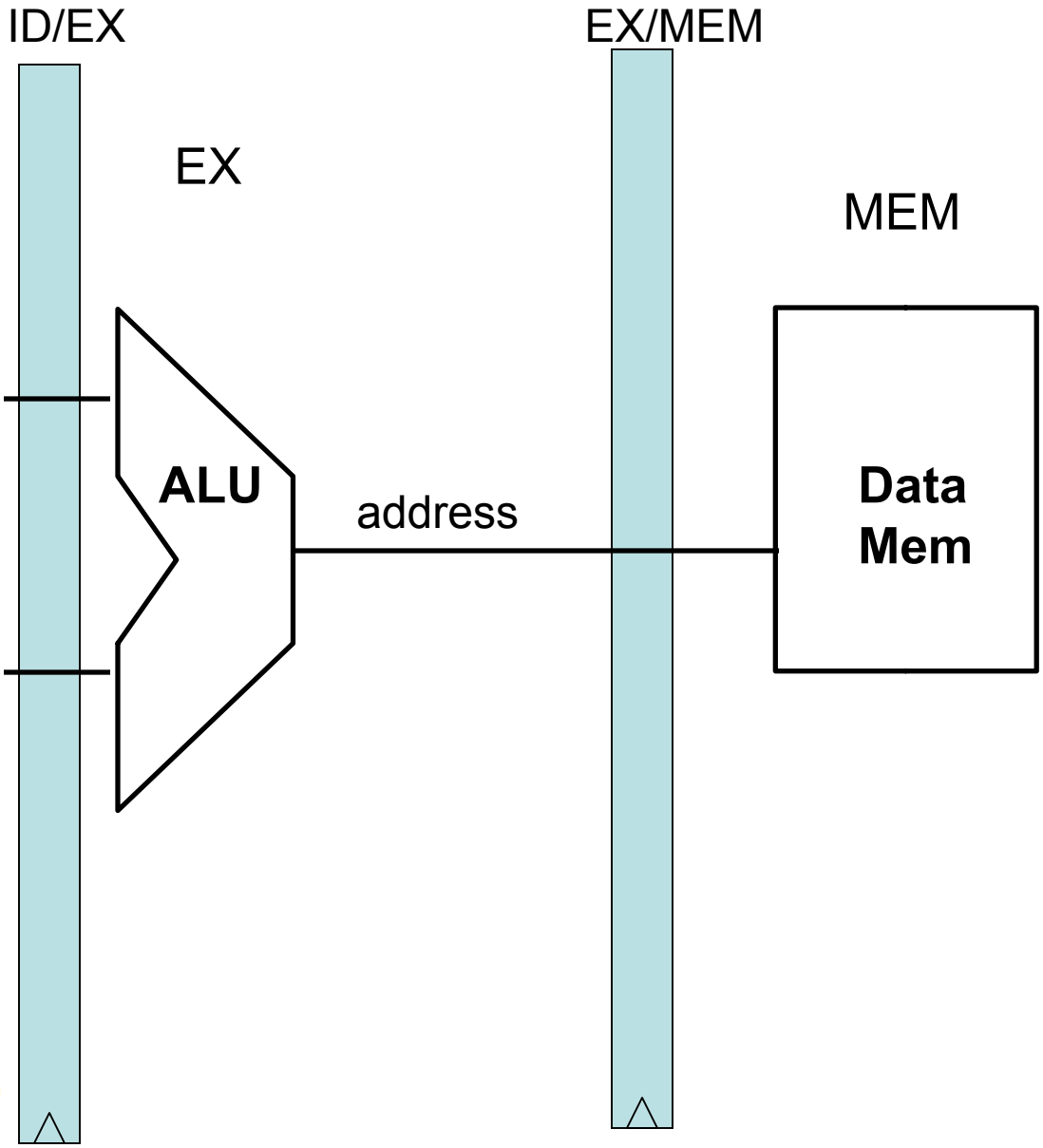


- The number following the break instruction (in this case 5), is placed into the Code segment of the instruction.
- You need to display the lower 7 bits of these numbers as part of the STAT signal during a break.
- Your processor should stall until it receives an external signal called “release.” Release is a debounced input from the board. What about simulation??

Synchronous Memory

- This lab uses synchronous memory.
  - Synchronous writes—just like last time
  - Synchronous reads—NOT LIKE LAST TIME
    - Address/control inputs to your memory have to be ready before the posedge of the cycle you want the results!
    - Why can't we use synchronous reads in the single-cycle processor?

Synchronous Memory

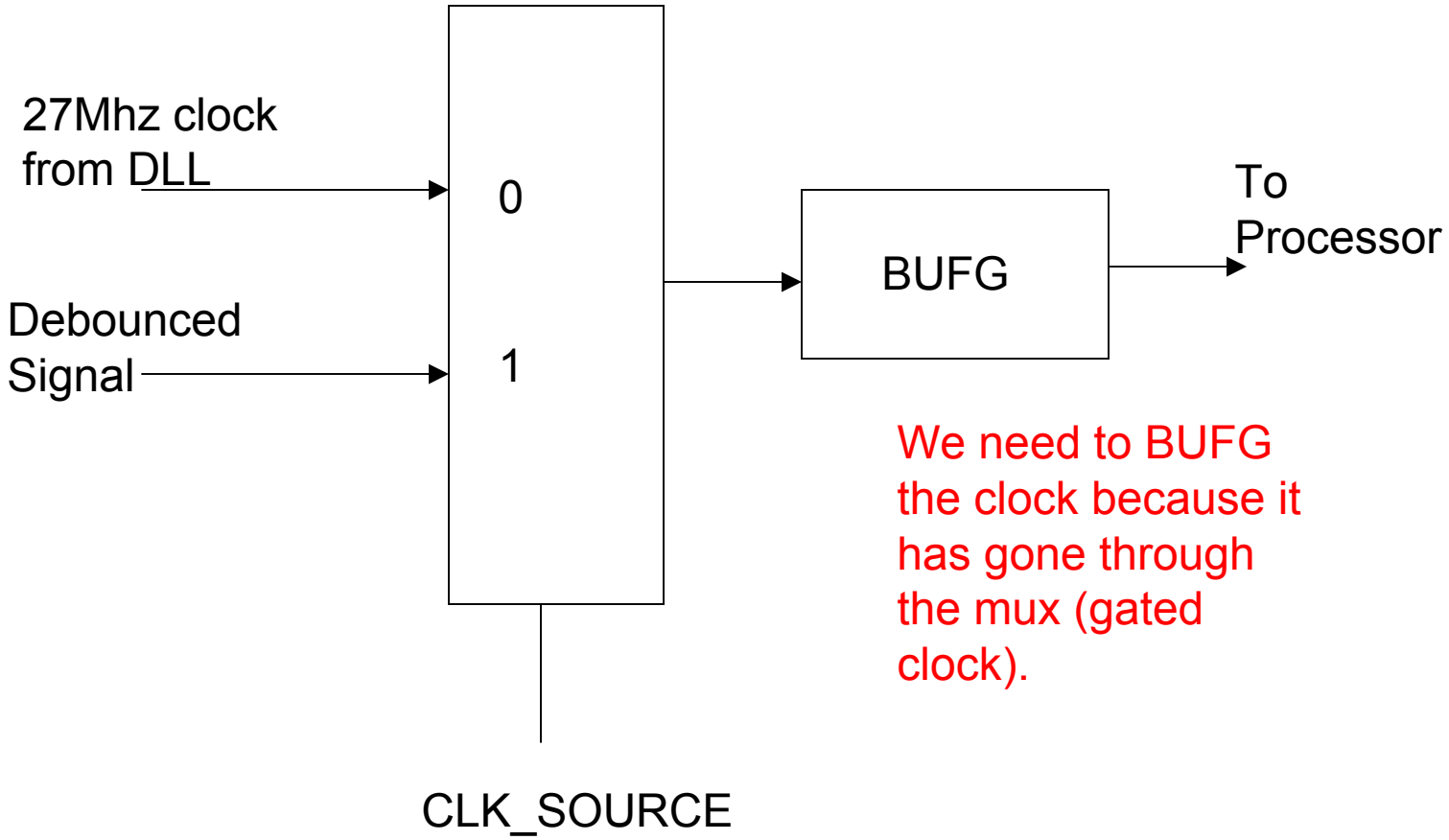


What happens is we pass the address through the pipeline register?

Does a synchronous datamem get the right result?

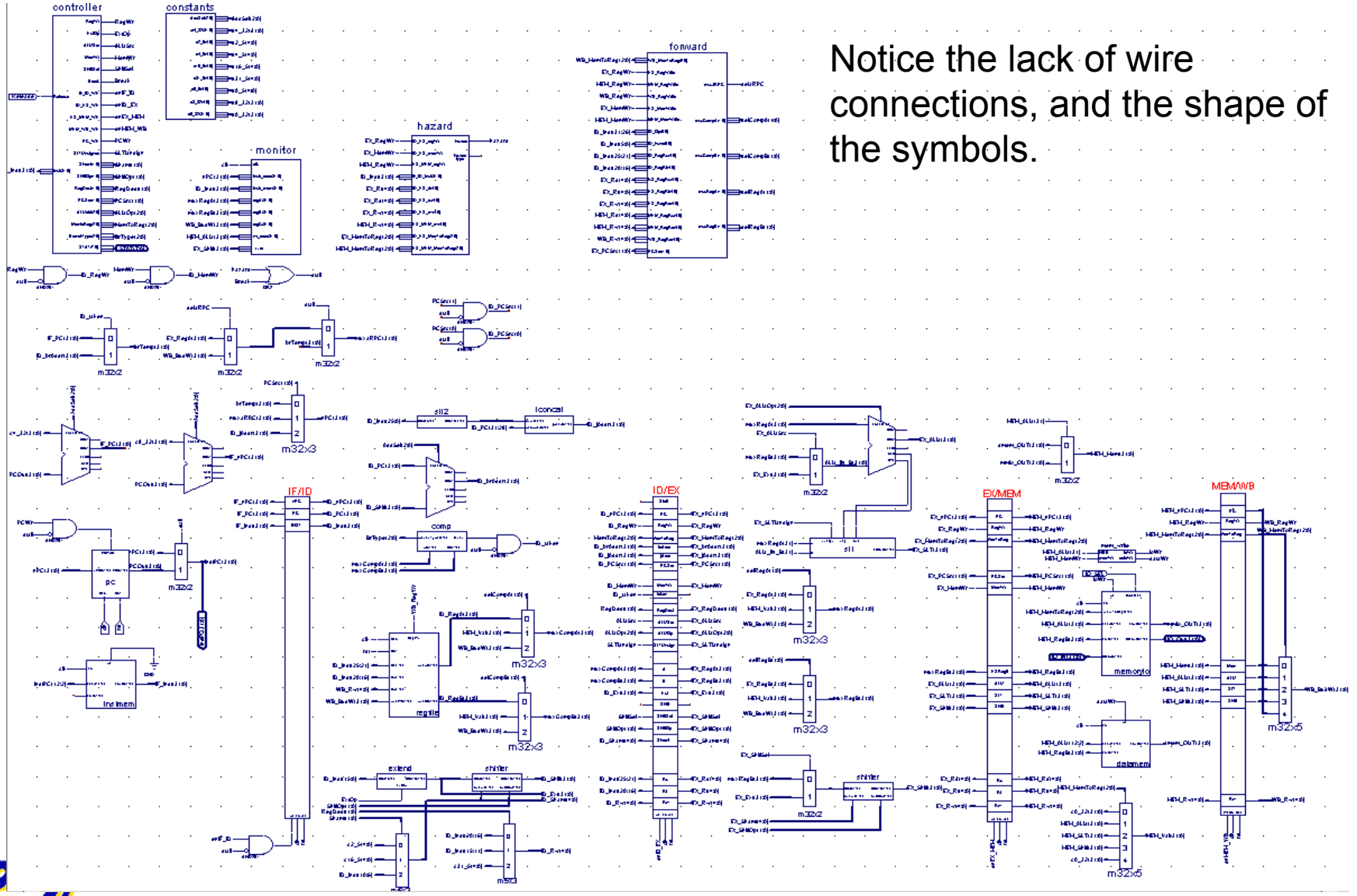
If the address is produced in cycle 3, what gets latched where in cycle 4?

Single Stepped Clock



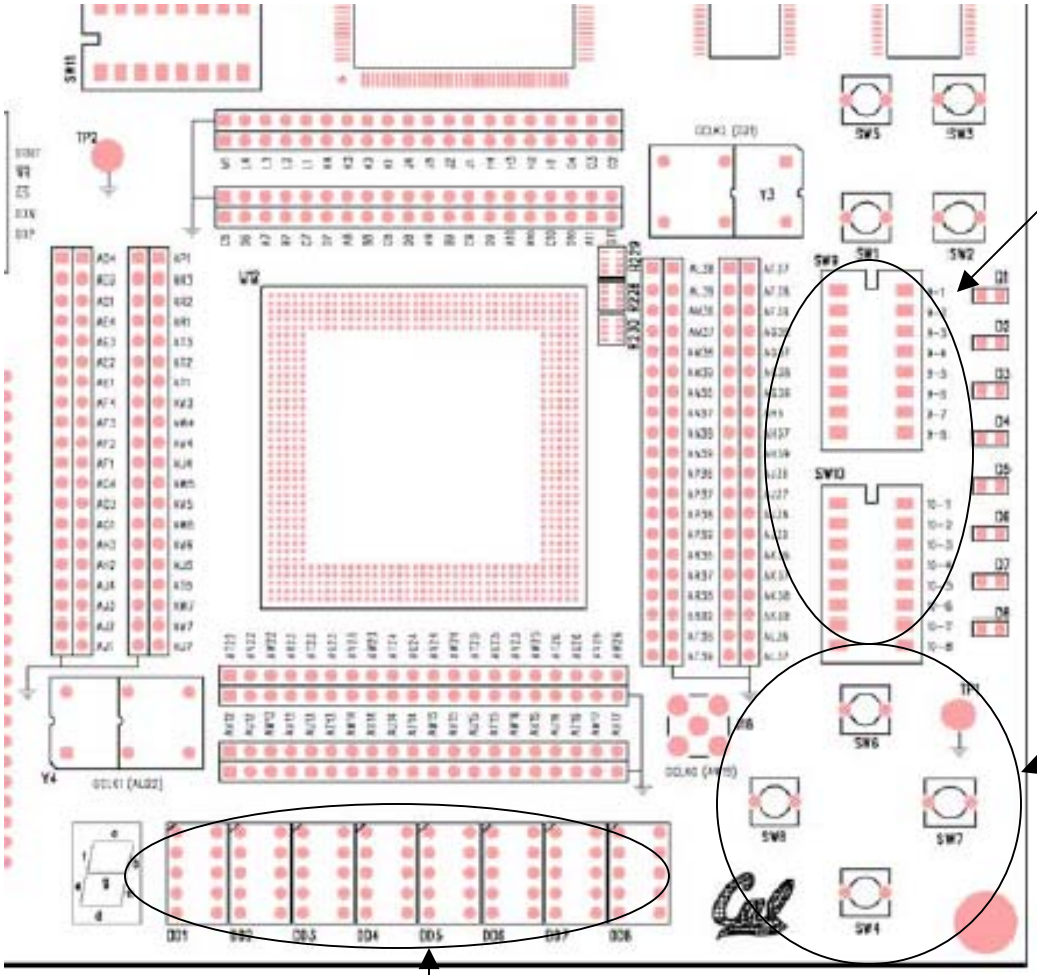
We need to BUFG the clock because it has gone through the mux (gated clock).

Schematics



Notice the lack of wire connections, and the shape of the symbols.

Board Connections



Input to IO, clock selector

STAT signals

Reset, single step, release, IO select

HEX LEDs