

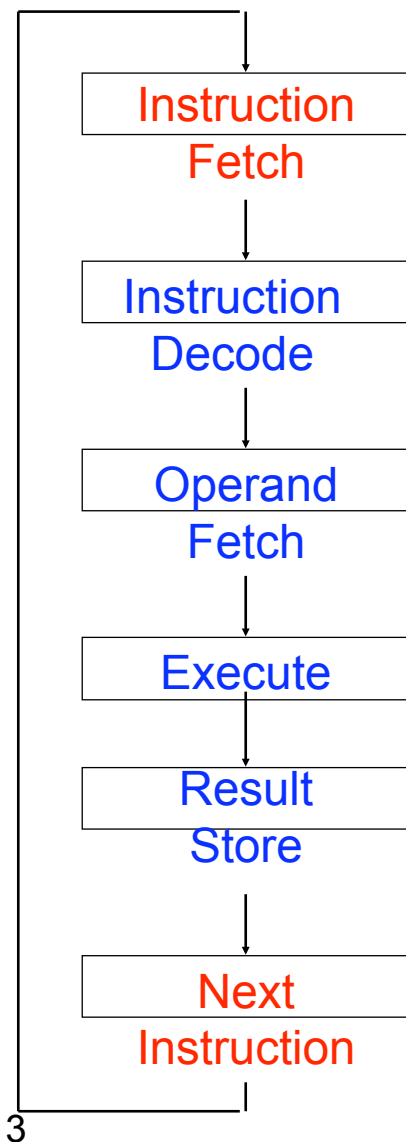
Administrivia

- Joe's lab hours
 - Monday 6-8, Tuesday 6-9, Thursday 6-9
- Any problems with lab 1?

Fetch Unit

CSE141L Lab 1b

The Instruction Execution Cycle



Obtain instruction from program storage

Determine required actions and instruction size

Locate and obtain operand data

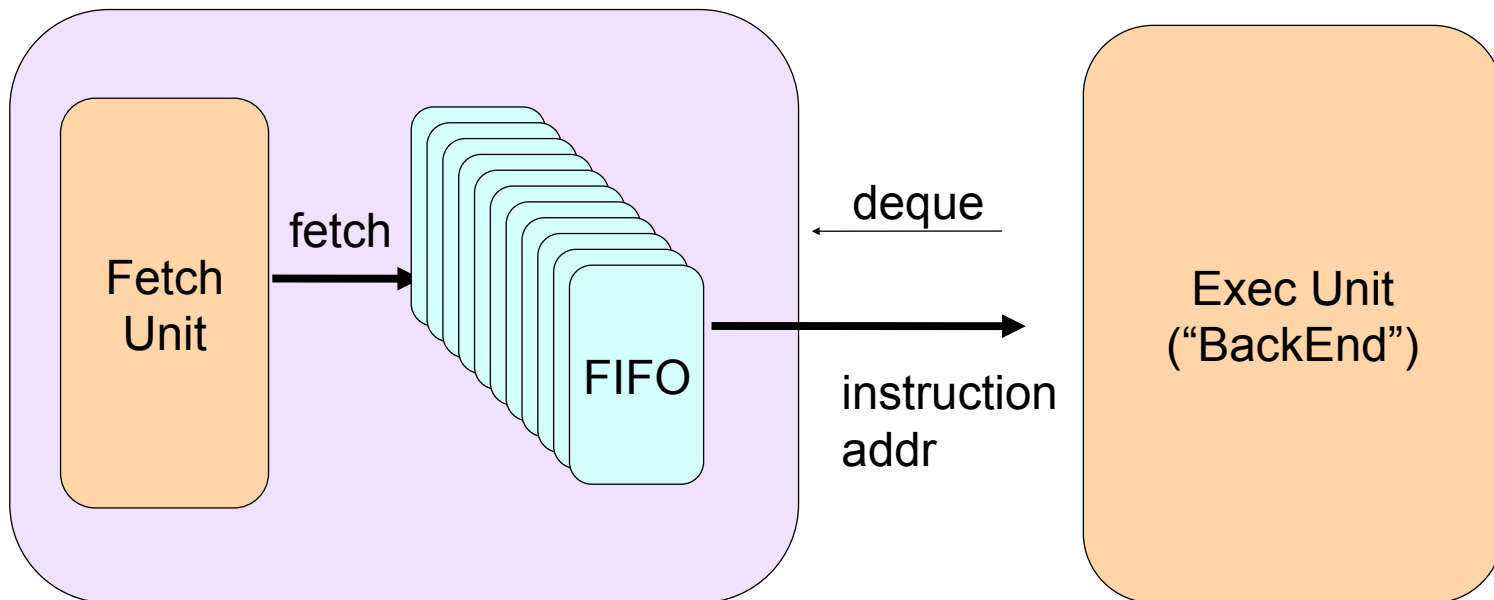
Compute result value or status

Deposit results in storage for later use

Determine successor instruction

Fetch Unit?

- Fetch instructions from I-mem
- Supply instructions to the Exec Unit
- Why use a FIFO?



Fetch Unit Roles

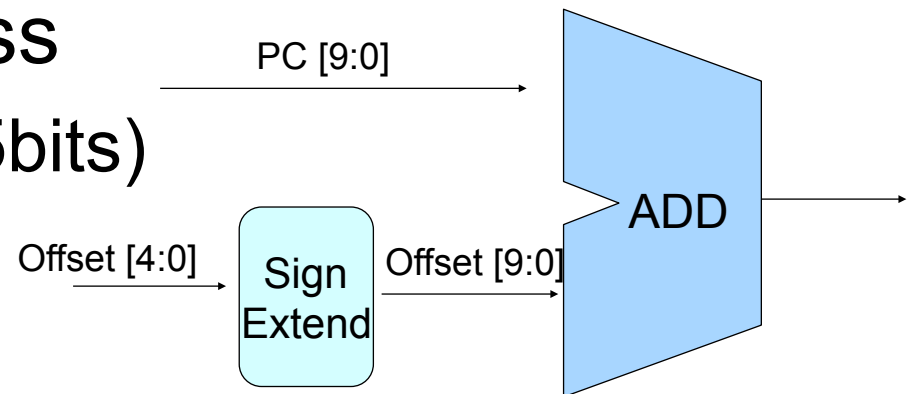
- Supply instructions to the exec unit
- Service I-Mem operations
 - load an instruction
 - store an instruction
- Branch Prediction
 - if (cond) do A, else do B
 - fetch A or B?
 - Instead of waiting the result of condition evaluation, predict where to go
 - For a misprediction, a recovery mechanism required

Simple Static Branch Prediction

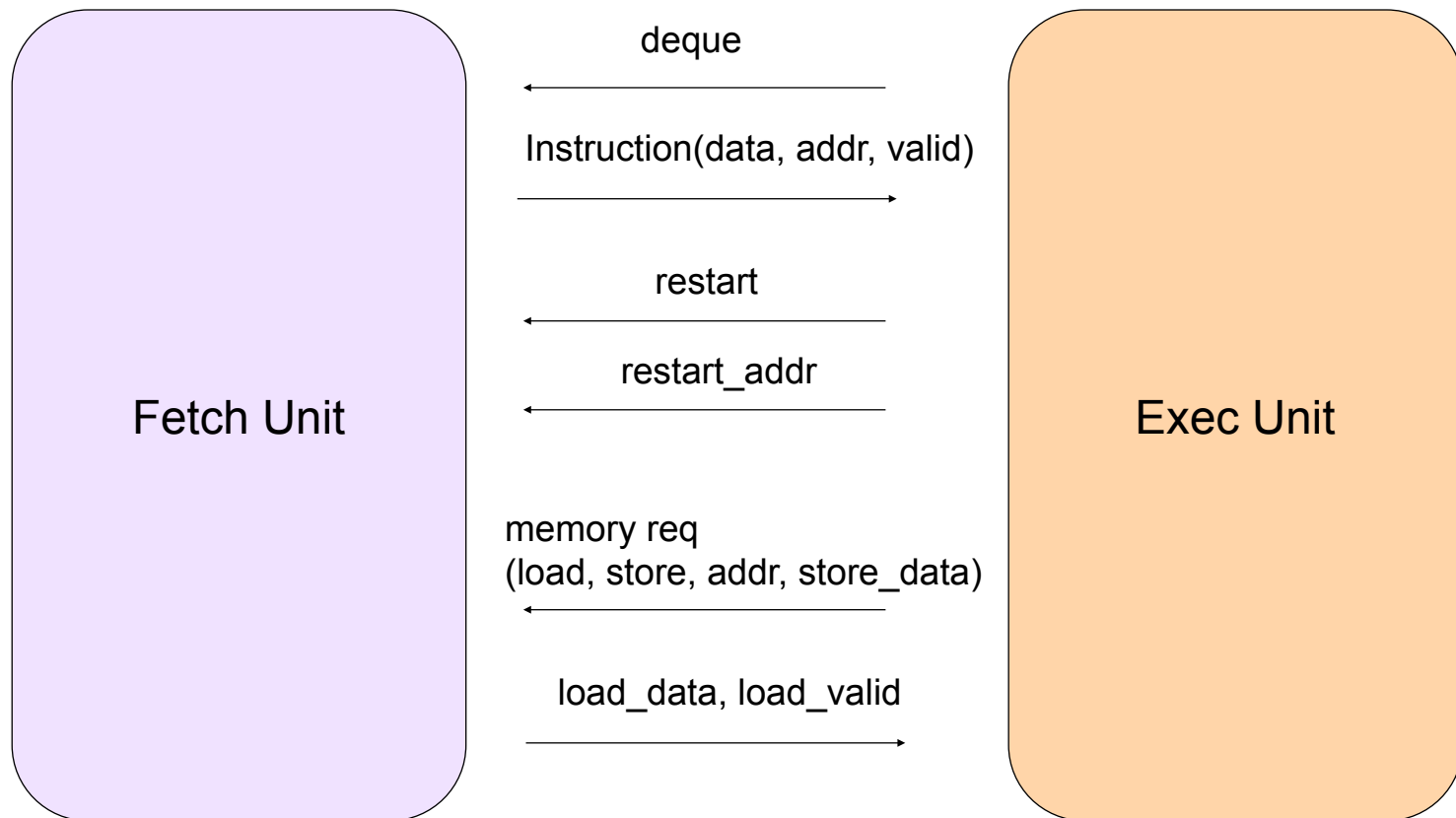


- P(1bit) - prediction bit
 - 0: not a branch, or a branch predicted untaken
 - 1: a branch, and predicted taken

- Branch target address
 - $PC(10\text{bits}) + \text{Offset}(5\text{bits})$



Fetch Unit Interface



```
module fetch #(parameter I_WIDTH = 17, A_WIDTH = 10, O_WIDTH= 5)
(
    input clk, reset,

    // normal operation, signals from the backend end
    input deque,
    input restart,
    input [A_WIDTH-1:0] restart_addr,

    // normal operation, signals from the front end to backend
    output [I_WIDTH-1:0] instruction_data,
    output [A_WIDTH-1:0] instruction_addr,
    output instruction_valid,

    input [A_WIDTH-1:0] load_store_addr,
    input [I_WIDTH-1:0] store_data,
    output [I_WIDTH-1:0] load_data,
    output load_valid,
    input load_store_1_0;
    input do_load_store;
)
```

Fetch Unit Operations

- deque
 - simple
- fetch
- restart
- memory
 - load
 - store

Operation - Fetch

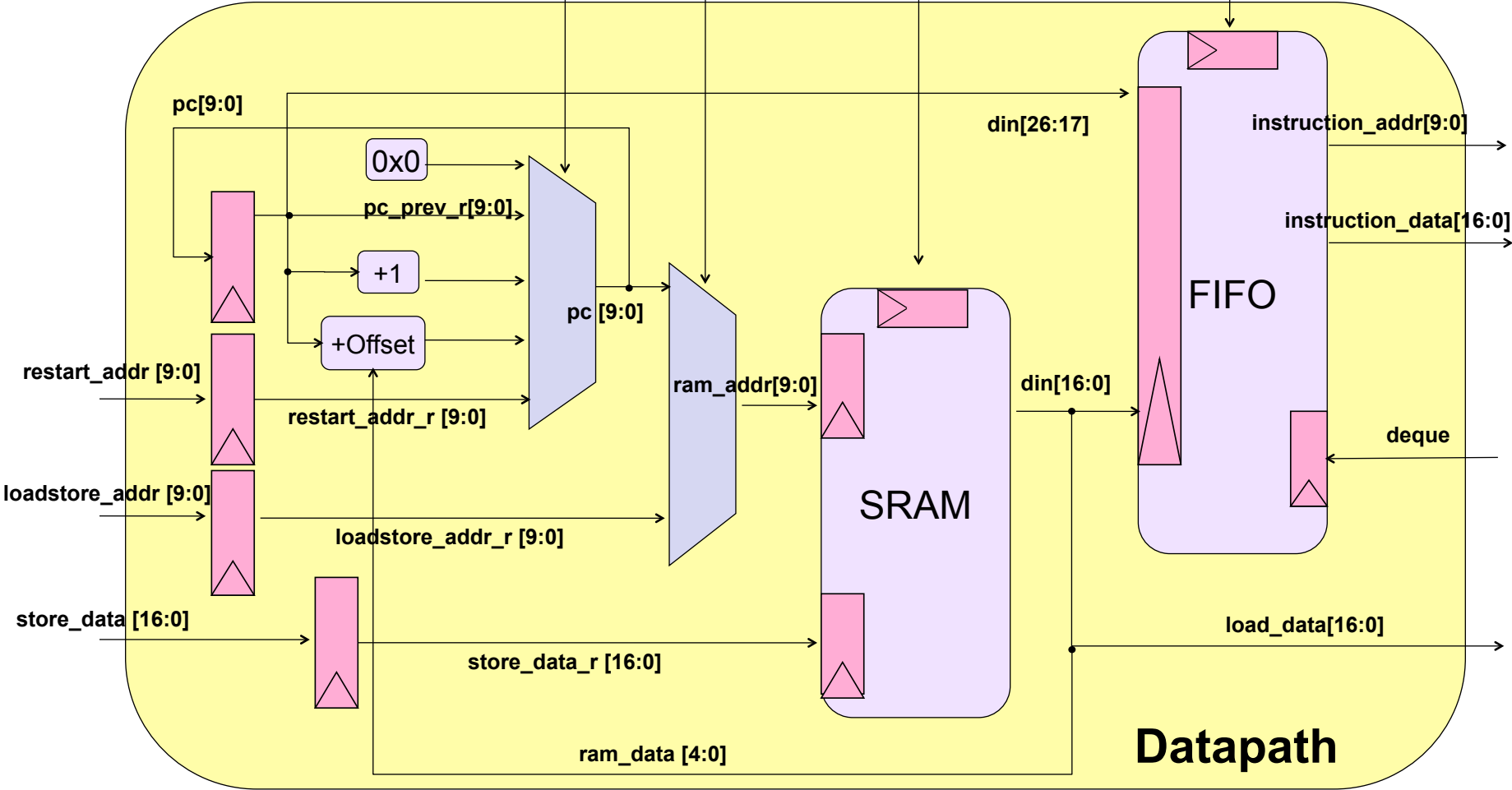
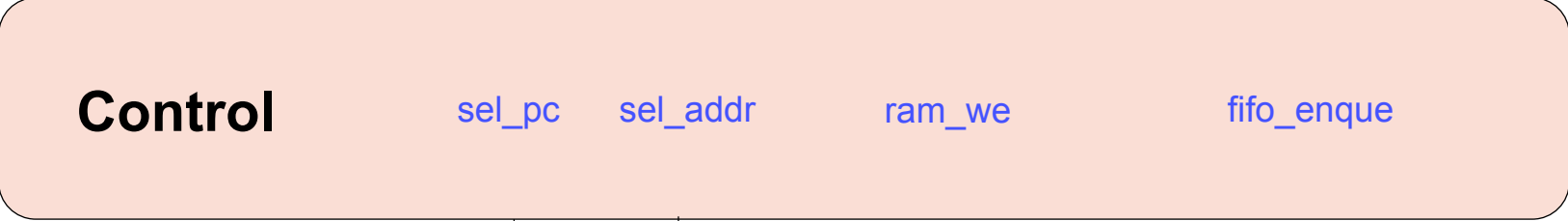
- If reset
 - Fetch the instruction at 0x0
- Else If (branch | branch_predicted)
 - Fetch the instruction at the branch target address
- Else
 - Fetch the instruction at {PC + 1}

Operation - Restart

- Why Restart?
- Operations
 - discard all the instructions in FIFO
 - Fetch the instruction at restart_addr
- Priority?
 - Reset > Restart > Normal Fetch

Operation - Memory

- Memory Operations
 - Load: read a data from the specified addr of the I-Mem
 - Store: write a data to the specified addr of the I-Mem
 - At most only one operation can be performed in a cycle
- Priority: why?
 - 1. load = store
 - 2. reset
 - 3. restart
 - 4. normal fetch



Lab 2

- Due next Friday
- Goal
 - Make the datapath Work
 - Understand the given datapath
 - Use given 2 module
 - RAM
 - FIFO
 - Implement 2 modules
 - Adder
 - Sign extender
 - Connect all things together
 - Synthesize, Implement, and analyze
 -
- Detailed instructions will be available tonight!