CS 3725 Feb. 28/12

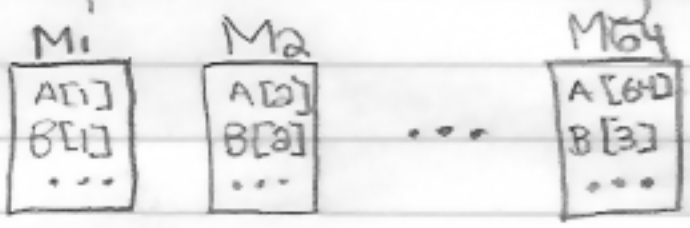
## Vector Processors / Array Processors

+ Vector Languages
+ Vectorizing Compiler

- In vector language
  ① $C \leftarrow A + B$ where A, B, C are declared vectors
  ② $C[1...100] \leftarrow A[1...100] + B[1...100]$

## Mapping Data into Memory Modules

a. 64 Independent Memory Modules




- So we can access A[1...100] in 2 memory cycles

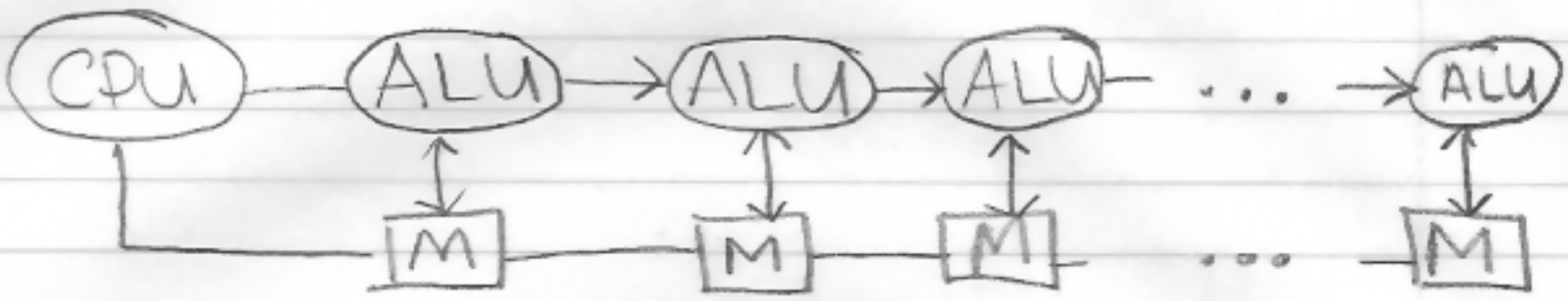
- 1 instruction fetch for $C \leftarrow A + B$

- If we have 64 ALU's (D-Units) each computing A[i] * B[i] in parallel, to compute 100 multiplication, we need 2 steps.

## Two Versions of SIMD Machines

### ① True SIMD (Vector) machines

- Abstract Diagram:




(i) CPU dispatches 1 instruction to all the ALUs

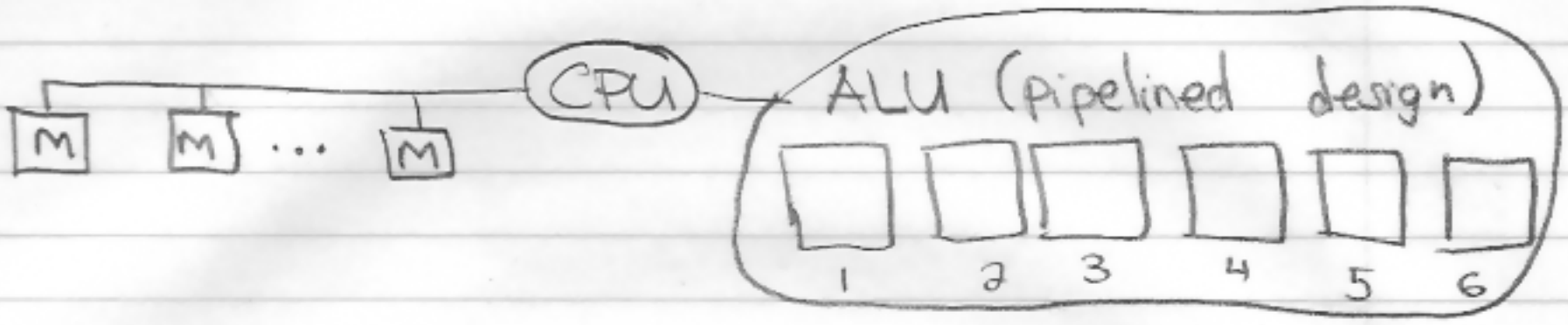
(ii) Each ALU is connected to its local memory which contains part of the data

(iii) ALU will perform the same instruction (dispatched by the CPU) onto its local data set

~ The connection between the (ALU-m) pairs can be
① Linear
② Cross Bar
③ Mesh
④ Butterfly
etc...

② Pipelined SIMD
(Example of Data Pipelining vs Instruction Pipelining [ex. MIPS])

• Abstract Diagram:




eg: ALU multiplier

1. Vector register read
2. Compare the exponents
3. Shift
4. Add
5. Normalize
6. Vector Register Write

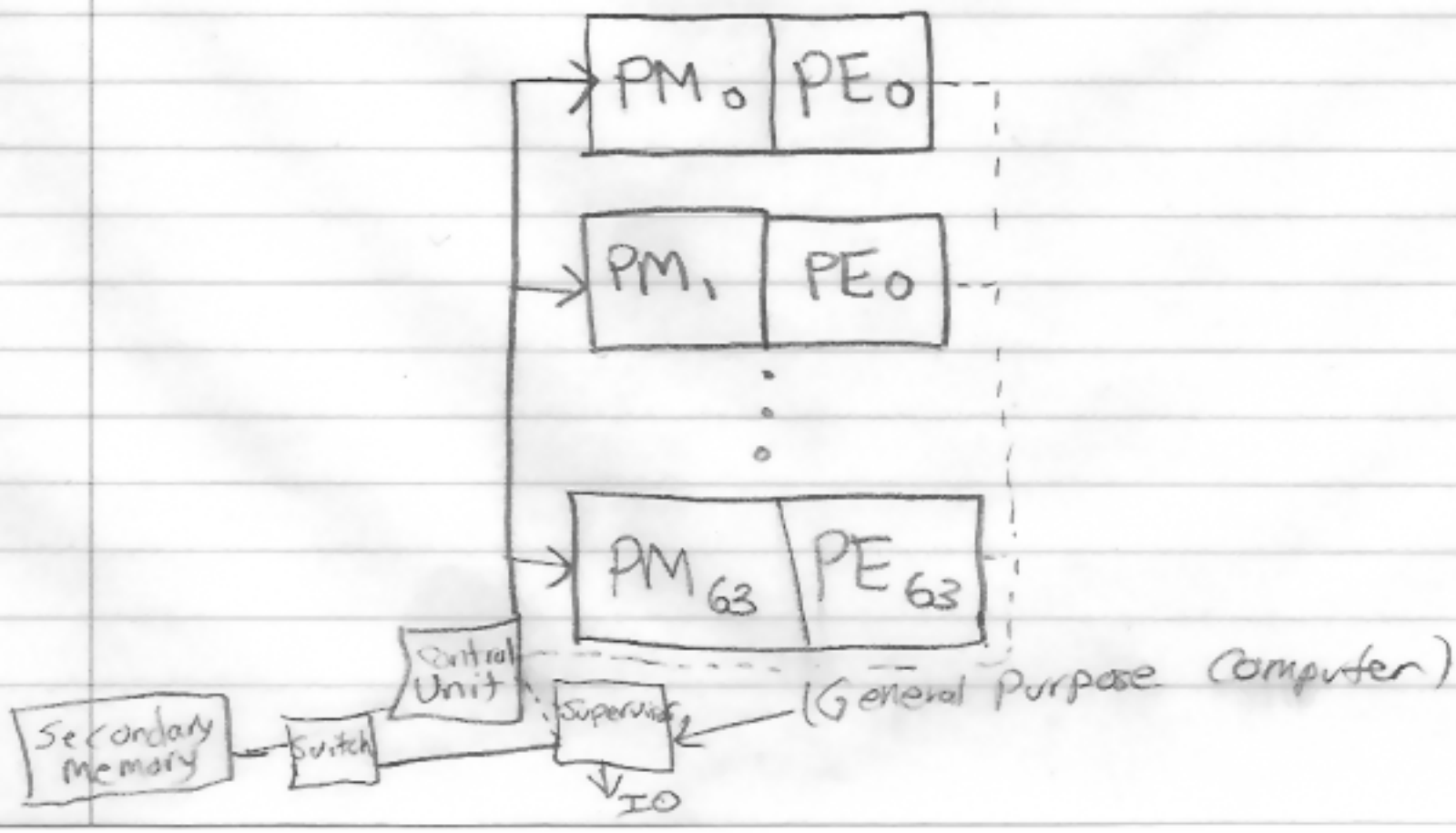
## Parallel SIMD Machines

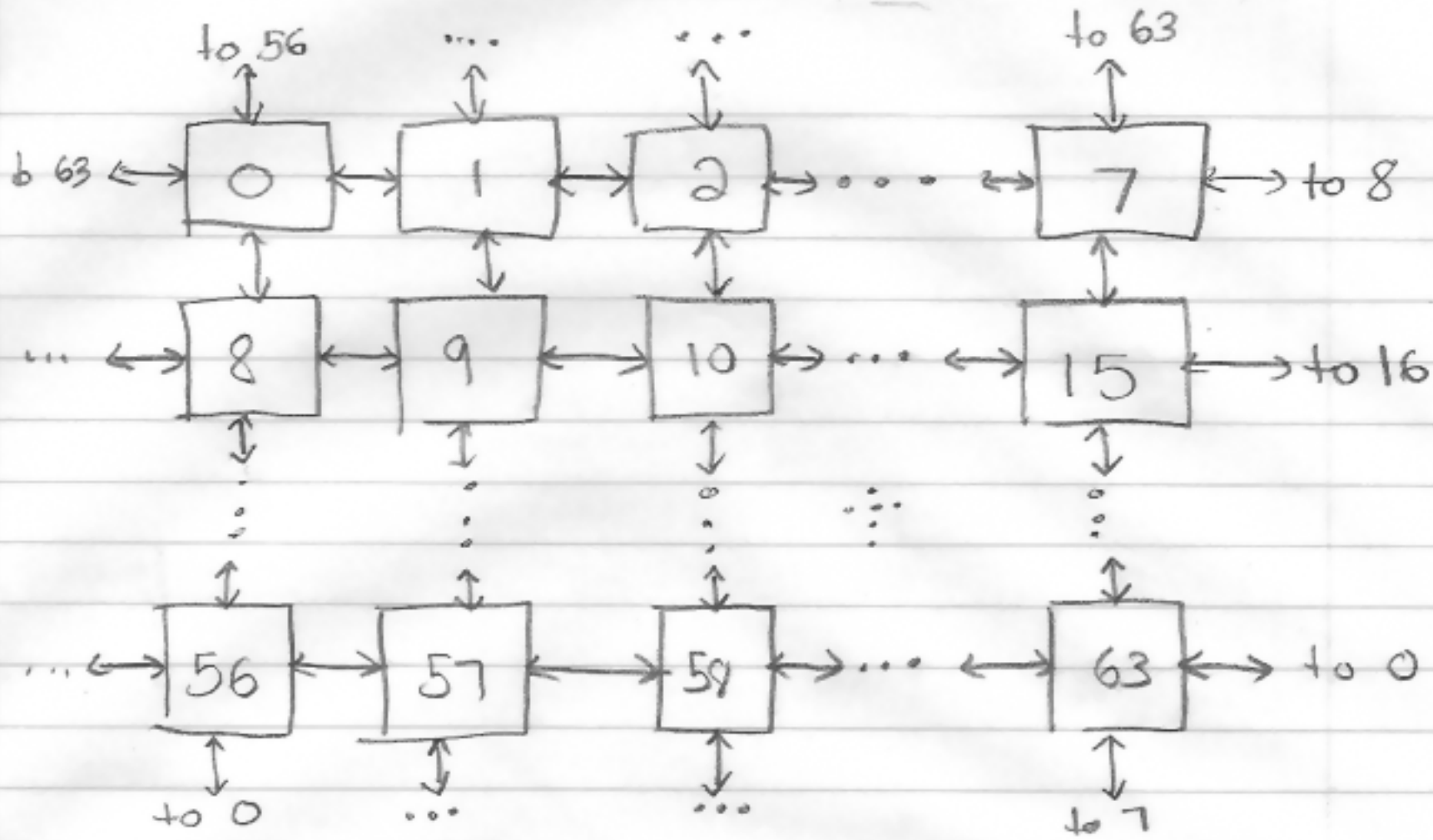
eg. ILLIAC IV

- 64 Processing Units (PU)
- Each PU = PE + PM

PE → Processing Element

PM → Processing Memory






- All 64 PU's are interconnected in 8×8 mesh networks
- Every PU is connected to its 4 nearest neighbors.

- 64 process elements perform the same instructions issued by the control units (ex. mult)

- Supervisor includes the usual
  - compiler
  - os
  - etc

- A PU can disable its self !!
  ① Control unit issues instruction asking PUs to set their mask vector according to a particular location

(ii) Then control unit issues the same instruction to all and asks PU's to perform if its mask vector allows it.