Shiraz University
Computer Science and Engineering Dept.
Computer Architecture Lab
Spring 2021
Instructor: Dr. M Raji
Teaching Assistants:
M Allahakbari, A Baghi, A Tavakoli, E Moeini, R Roshan

Final Project

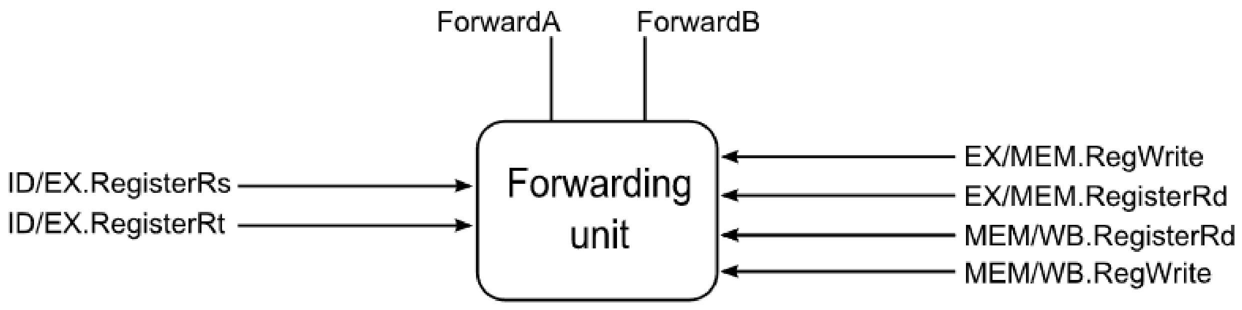
The addition of pipeline to MIPS architecture can significantly improve the throughput of the processor. However, adding pipelining introduces a few problems known as hazards (which you have seen in previous courses).

In this project, we are interested in fixing data hazards. Data hazards can be categorized into two groups, register and memory data hazards. Register data hazards can be further categorized into the following types:

i. RAW (read-after-write)
ii. WAW (write-after-write)
iii. WAR (write-after-read)

Register data hazards can be handled with the help of a forwarding unit. The forwarding unit diagram is as follows:

(Implementation Logic of this unit can be found in your textbook)

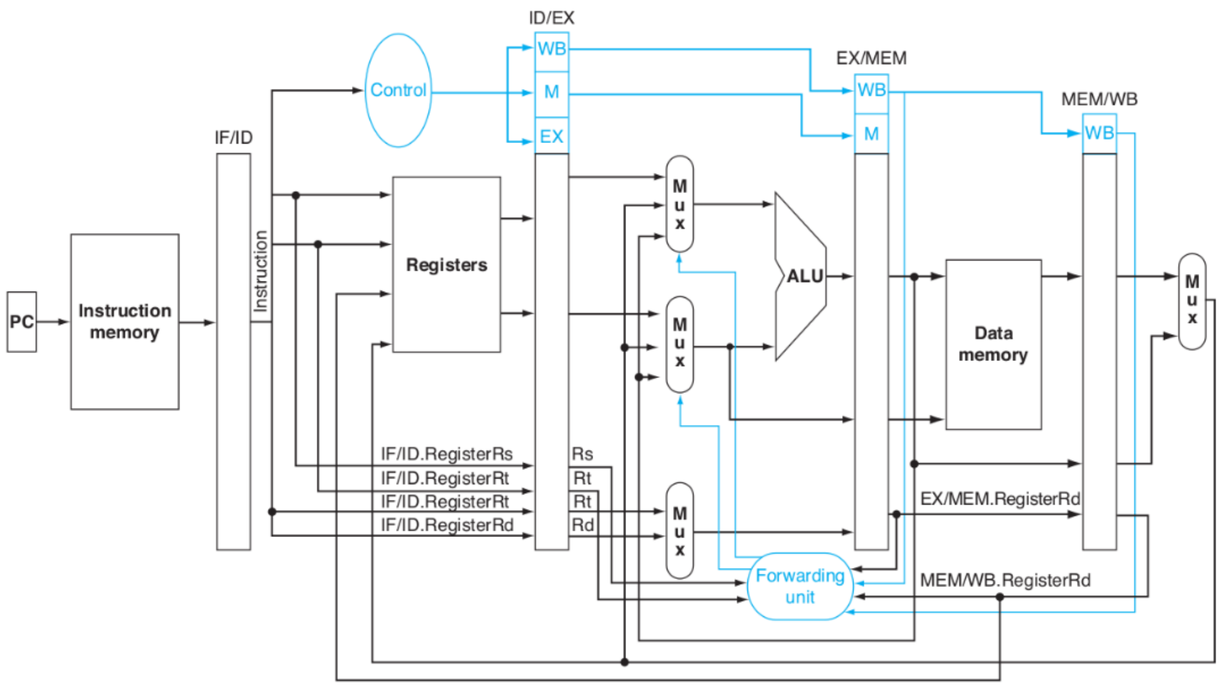


The input/output values are characterized as:

Input Value	Width	Description
EX/MEM.RegisterRd	5 bits	destination register for instruction k + 1
MEM/WB.RegisterRd	5 bits	destination register for instruction k
EX/MEM.RegWrite	1 bit	RegWrite setting for instruction k + 1
MEM/WB.RegWrite	1 bit	RegWrite setting for instruction k
ID/EX.RegisterRs	5 bits	left operand for instruction k + 2
ID/EX.RegisterRt	5 bits	right operand for instruction k + 2

Output Value	Width	Description
ForwardA	2 bits	controls selection of left operand to ALU
ForwardB	2 bits	controls selection of right operand to ALU

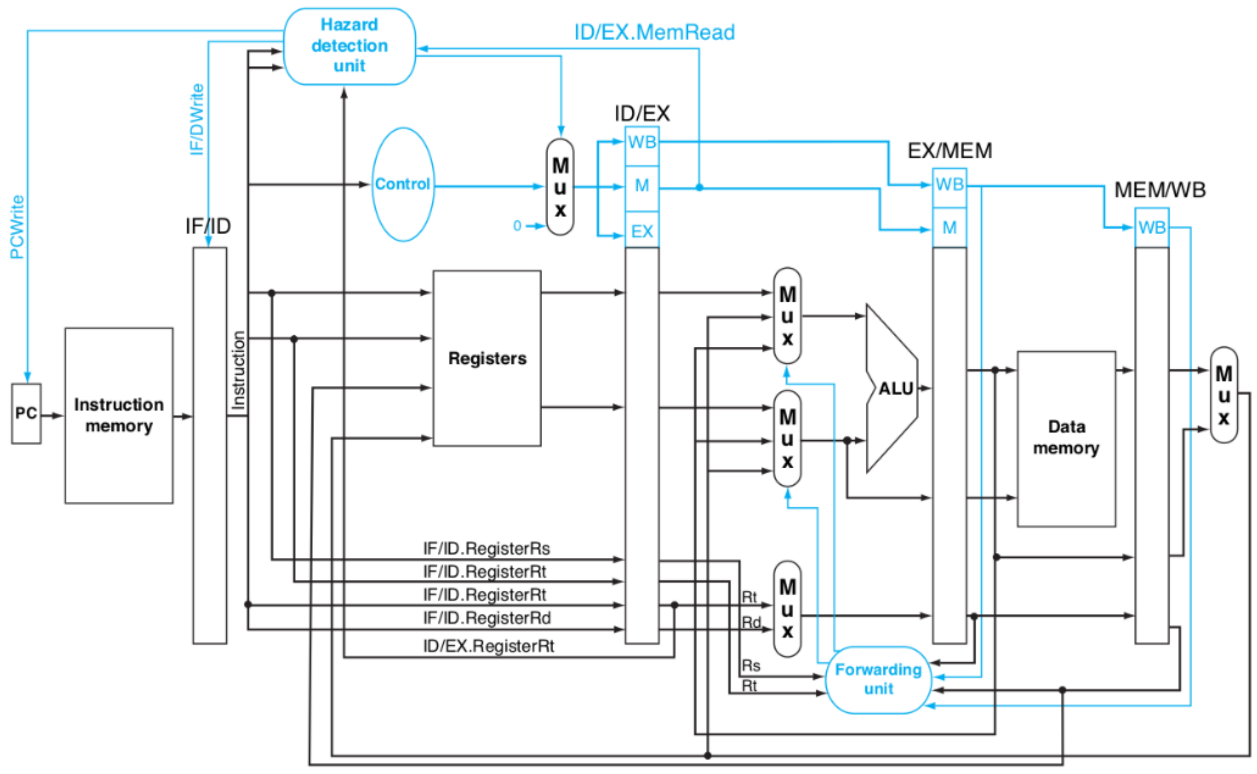
The forwarding unit should be placed in the MIPS pipeline in the following manner:



The next data hazard that should be handled is memory hazard. Forwarding unit cannot handle this type of hazard for memory and we need to implement another unit called the hazard detection unit. This unit is responsible for adding bubbles between subsequent instructions to memory write that try to read from the same location. The logic behind this unit is as follows:

```
if (ID/EX.MemRead and
   ((ID/EX.RegisterRt = IF/ID.RegisterRs) or
     (ID/EX.RegisterRt = IF/ID.RegisterRt)))
      stall the pipeline
```

The placement of this unit is shown below:


Hazard detection unit
ID/EX.MemRead
IF/DWrite
PCWrite
ID/EX
EX/MEM
MEM/WB
IF/ID
WB
M
EX
Control
Mux
0
PC
Instruction memory
Instruction
Registers
ALU
Data memory
IF/ID.RegisterRs
IF/ID.RegisterRt
IF/ID.RegisterRt
IF/ID.RegisterRd
ID/EX.RegisterRt
Rt
Rd
Rs
Rt
Forwarding unit