Single Cycle RISC-V datapath and Control

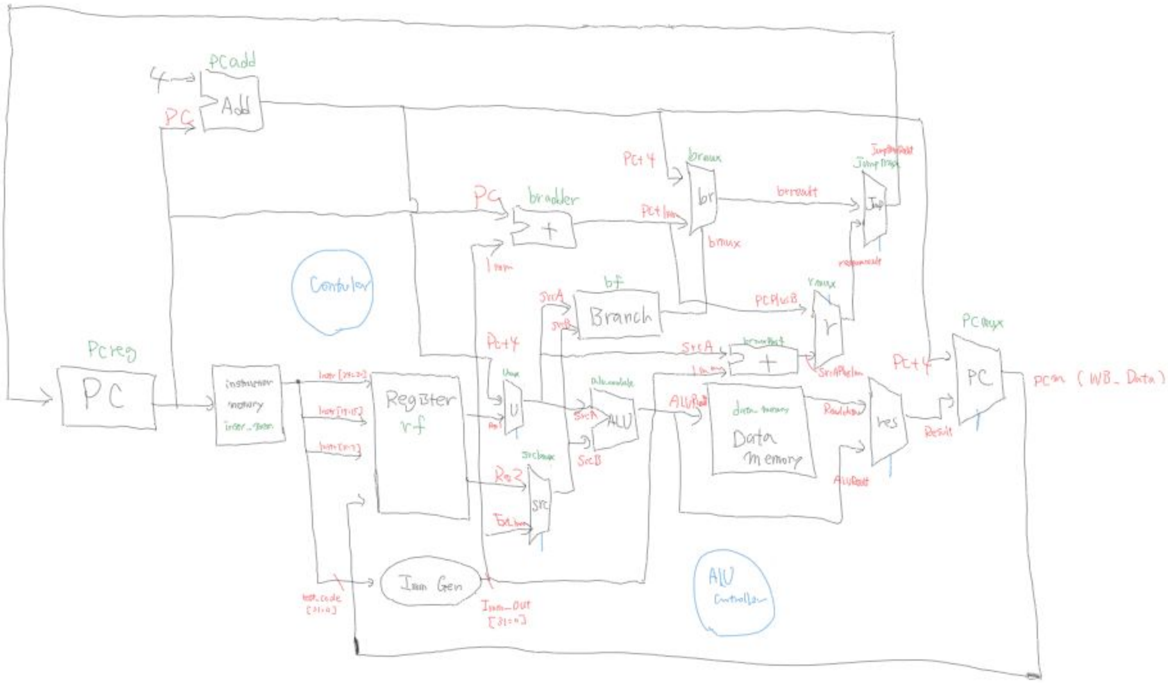
University of California Irvine

28763963 Yuki Hayashi
79364141 David Tiao

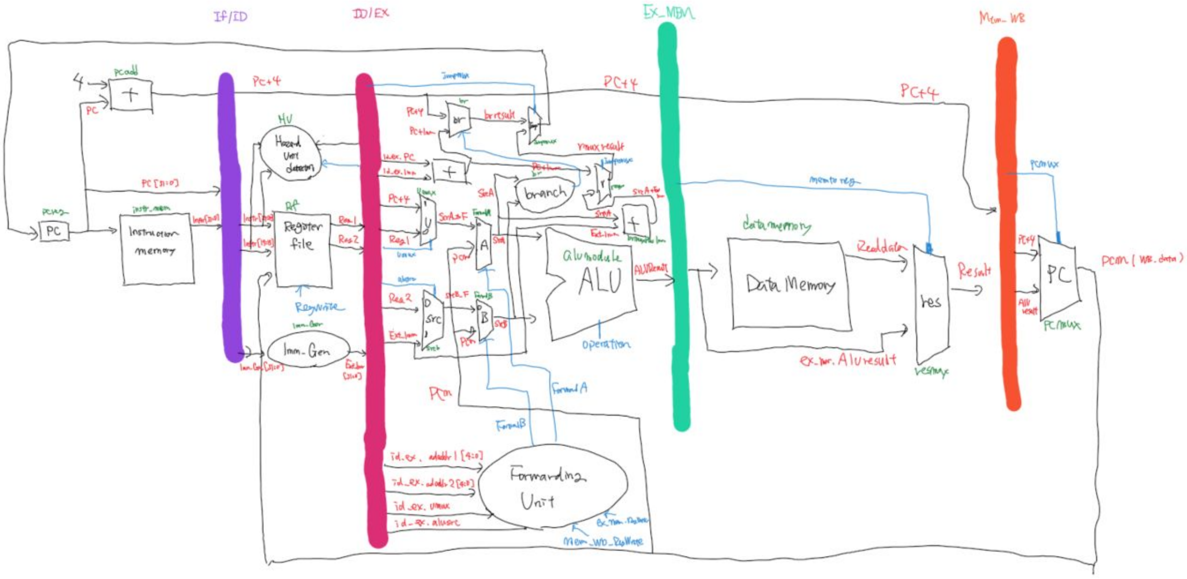
Introduction

The goal of this assignment is 1) to implement the five stages pipelining in RISC-V, IF(Instruction fetch), ID(Instruction decode), EXE(Execution), MEM(Memory), and WB(Write Back), 2) to implement forwarding unit and hazard detection unit.

Modified RISC-V



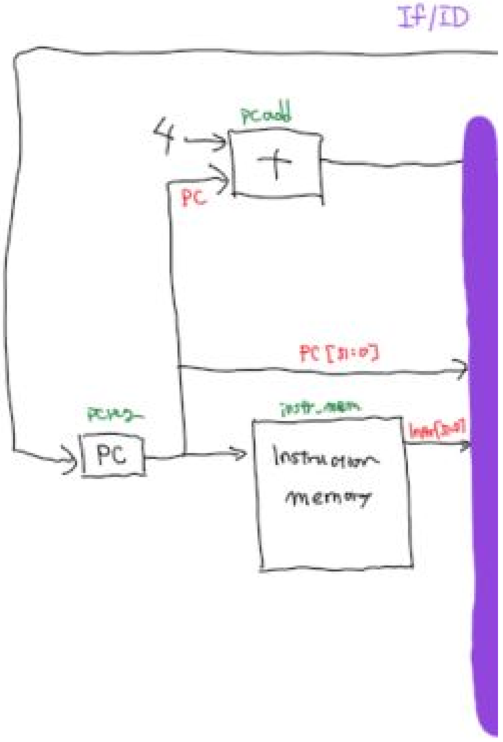
Pipelined Version



Block Diagram & Explanations

1. Single Cycle versus Pipeline

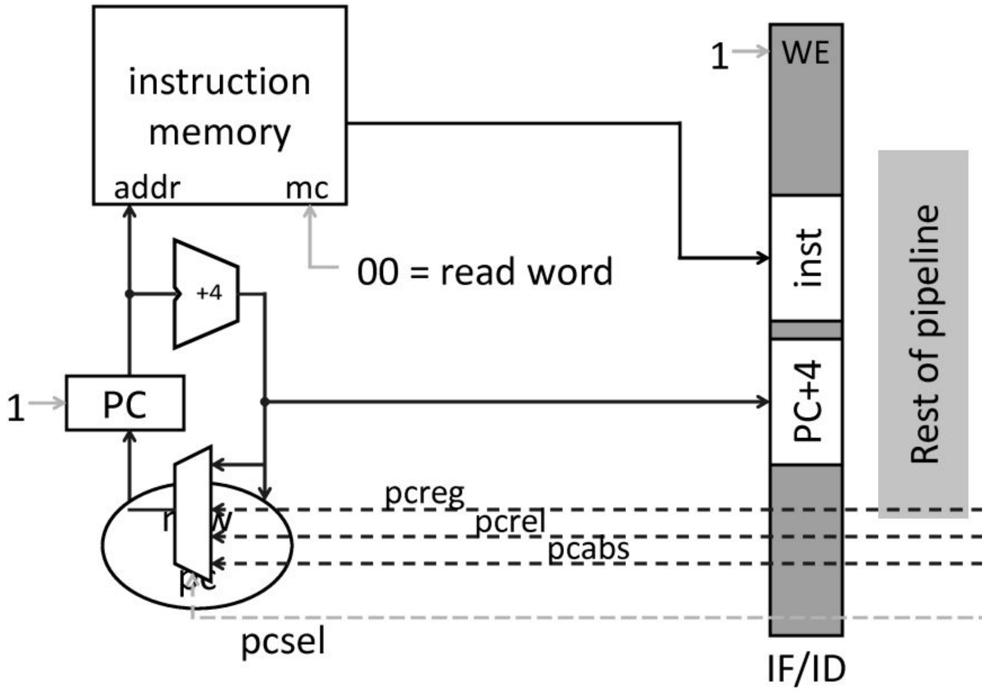
Instruction Fetch and Instruction Decode (IF_ID)



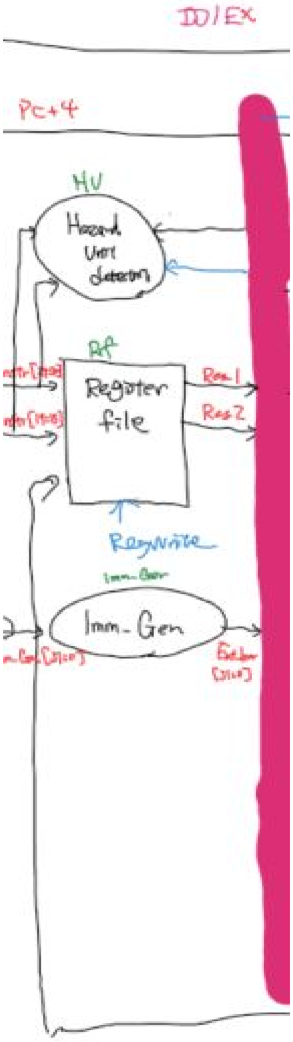
IF_ID

In the first stage of the pipeline, Instruction Fetch grabs the current program counter and increments it at the end of the cycle. It then writes the instruction bits for future decoding and adds 4 to the counter for computing branch targets. These two things are the values of interest that get written to the IF/ID block.

In other words, the program counter is sent to the instruction cache in order to read the current instruction. At the same time, the PC predictor predicts the address of the next instruction by incrementing the PC by 4. After this fetch is complete, the state is written to the pipeline register. The next stage will read from this IF/ID register.

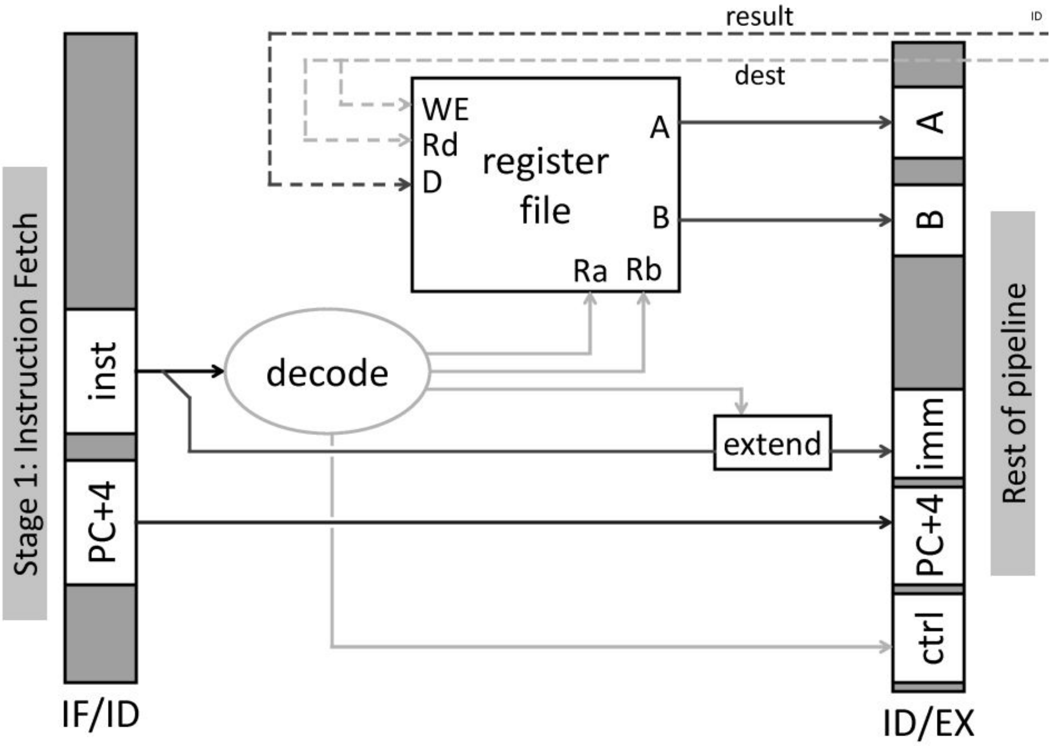


Instruction Decode and Execution (ID_EX)

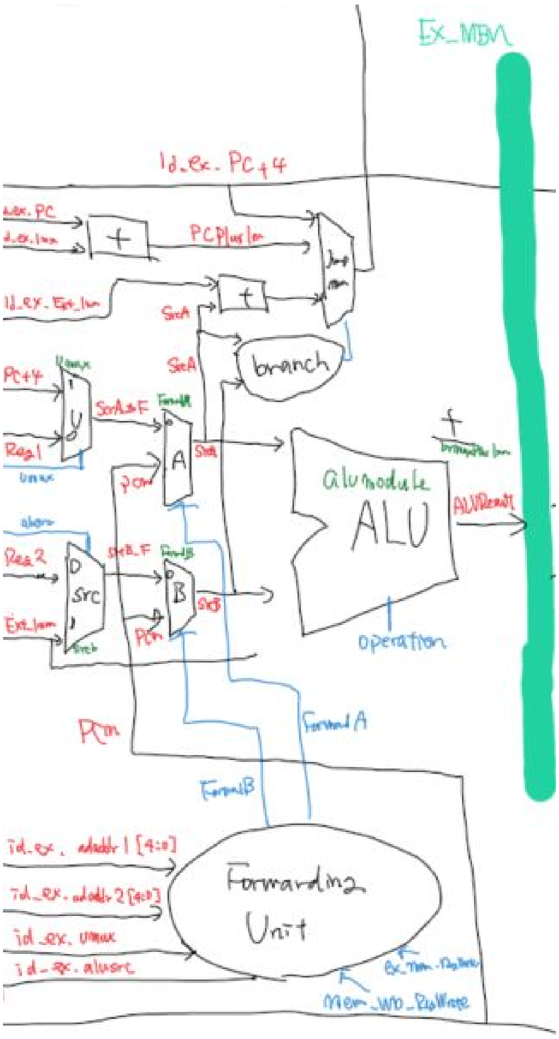


ID_EX

In this second stage of the pipeline, the OpCode bits are read and decided in order to prepare the control signal for future stages. From the register file, the input operands are read based on the decoded instruction bits. After this is complete, the processor writes the current state to the ID/EX pipeline register. This includes the OpCode, register contents, immediate operand, control signals from the instruction, destination register, and the program counter plus 1.

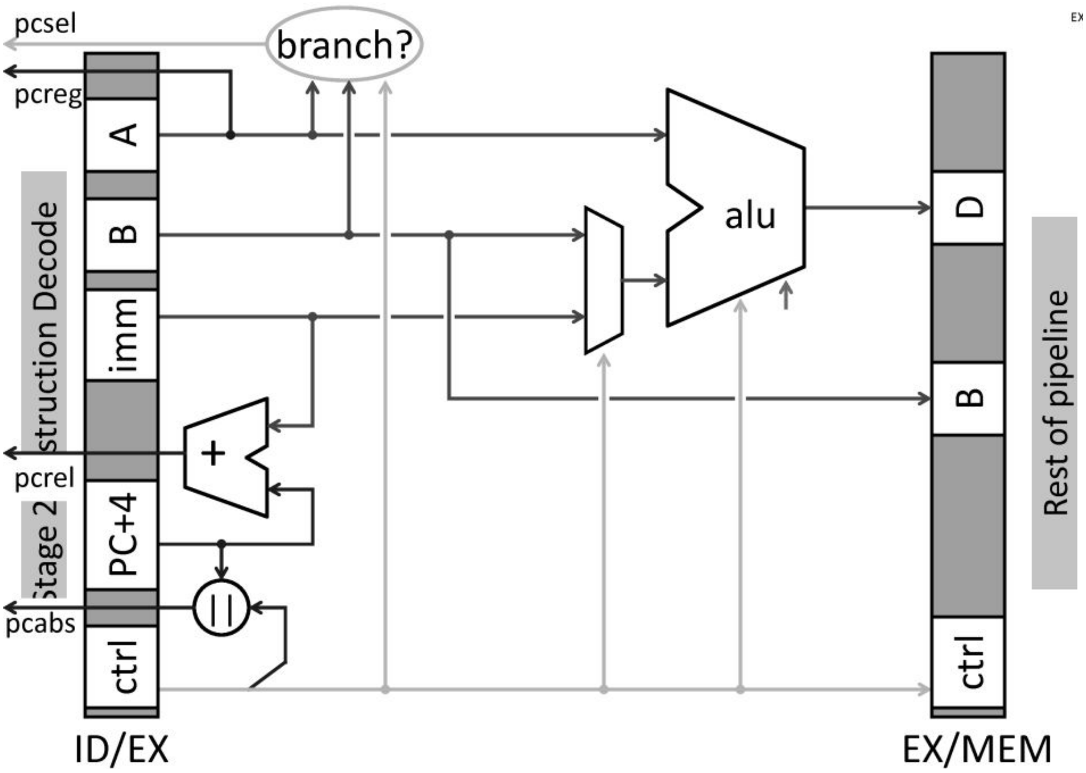


Execute and Memory Access (EX_MEM)

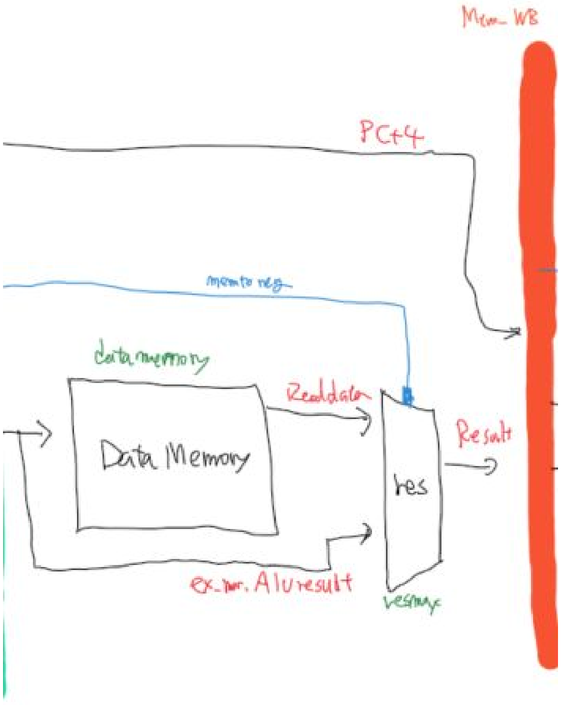


EX_MEM

During the execution step, all the ALU operations are performed. Here, the instruction's result is computed; the control signals that are passed in from the ID/EX pipeline register provide the proper control bits to select which operations to execute, and srcA + srcB are used at inputs to the ALU. Through a series of different multiplexers, the proper data is selected depending on the other control bits connected to the muxes in question. As this is happening, the program counter relative branch target is also computed just in case that it is a branch instruction. The state of all this data is written to the EX/MEM pipeline register. It includes the ALU result, program counter offset plus one, and finally the control signal bits from the instruction itself. At the end of the cycle, the decision to branch or jump is made.

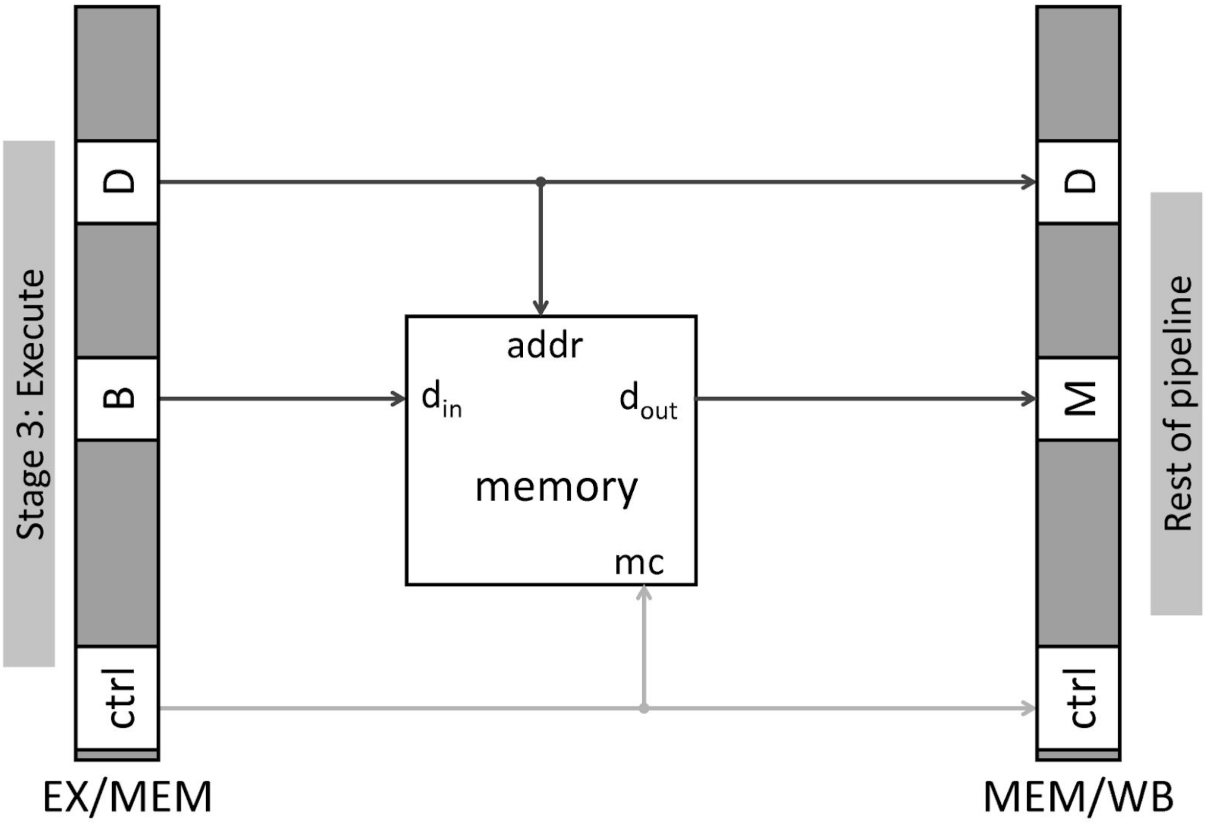


Memory Access and Writeback (MEM_WB)

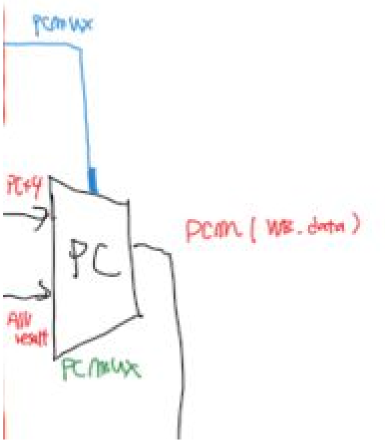


MEM_WB

During this stage at each cycle, the processor reads the EX/MEM pipeline register to fetch the required values and control bits. Afterwards, a memory store and/or load operation is performed if it is required. The result of this operation is the address of the ALU output. Once this has been complete, the data will be written to the MEM/WB pipeline register. This information includes: control information, memory operation result, ALU operation pass, and Register Index.

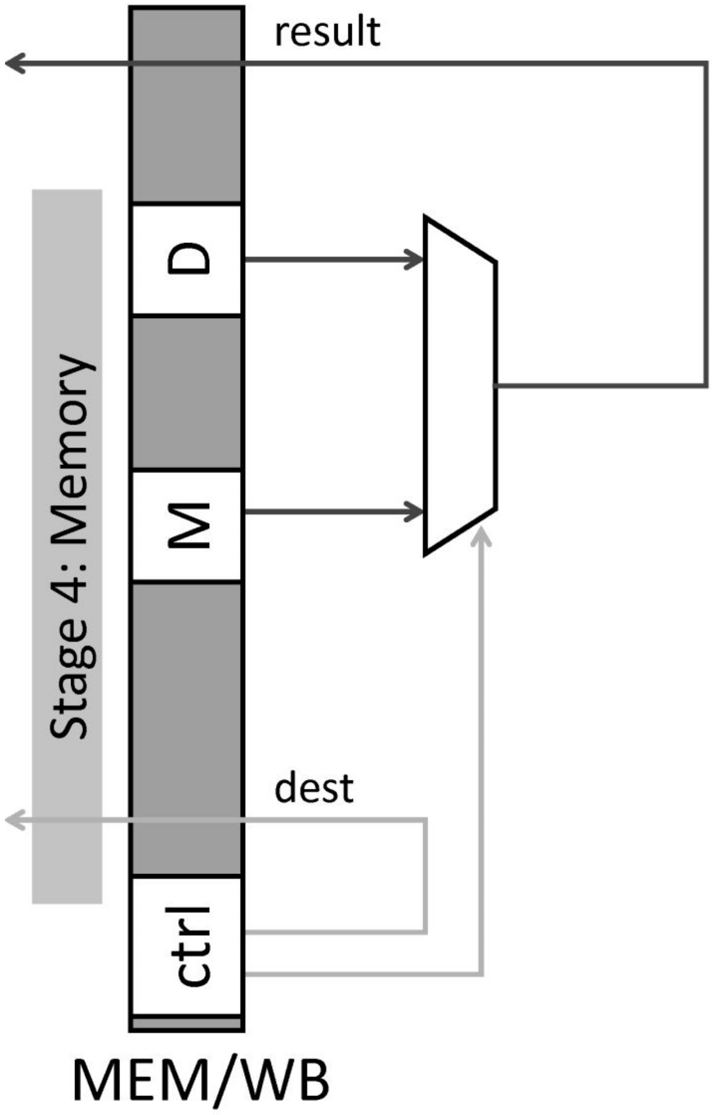


Write Back (WB)



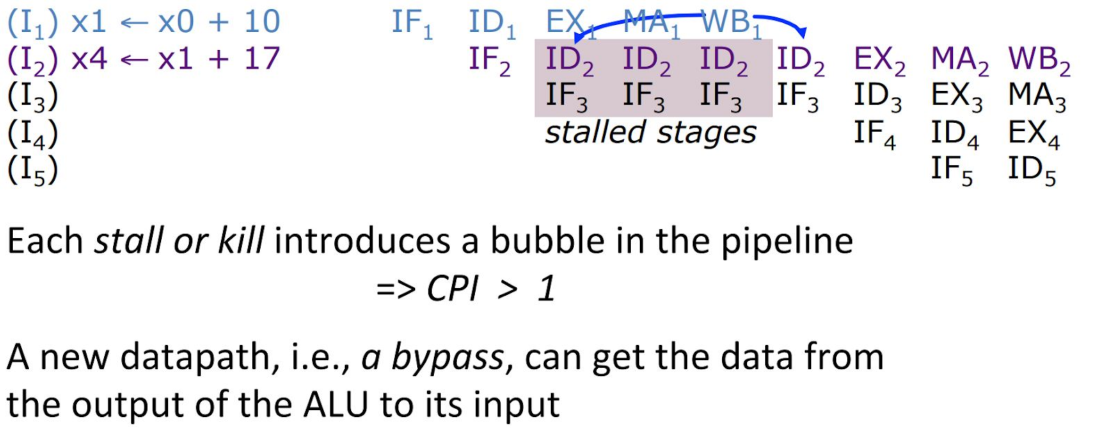
WB

For every cycle, this stage reads the MEM/WB pipeline register in order to retrieve data and control bits. From this, the processor is able to select the correct value and write it to the register file. For both single and double-cycle instructions, the results are also written to the register file.



2. Data Hazards

Data hazards occur when an instruction, scheduled blindly, would attempt to use data before the data is available in the register file. There are two main ways to avoid data hazards: Bypassing/Forwarding and Pipeline Interlocking. For our processor, we have implemented a forwarding unit to avoid these data hazards. The forwarding unit outputs control bits to two MUXs, srcA and srcB. Depending on the output of the control/select bits, the processor will choose what data to send to the ALU for operations to be performed.

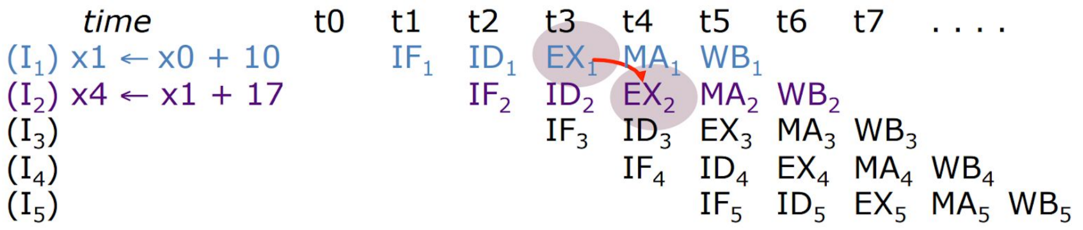


(I_1) $x1 \leftarrow x0 + 10$	IF_1	ID_1	EX_1	MA_1	WB_1				
(I_2) $x4 \leftarrow x1 + 17$		IF_2	ID_2	ID_2	ID_2	ID_2	EX_2	MA_2	WB_2
(I_3)			IF_3	IF_3	IF_3	IF_3	ID_3	EX_3	MA_3
(I_4)			*stalled stages*				IF_4	ID_4	EX_4
(I_5)								IF_5	ID_5

Each *stall or kill* introduces a bubble in the pipeline

=> *CPI > 1*

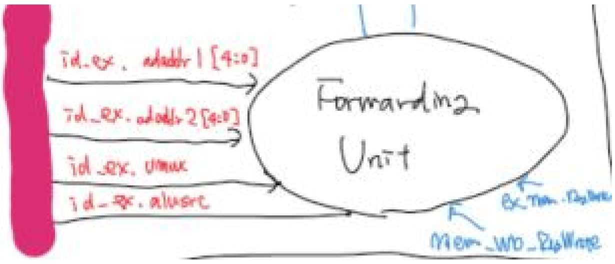
A new datapath, i.e., *a bypass*, can get the data from the output of the ALU to its input



time	t0	t1	t2	t3	t4	t5	t6	t7	. . . .	
(I_1) $x1 \leftarrow x0 + 10$		IF_1	ID_1	EX_1	MA_1	WB_1				
(I_2) $x4 \leftarrow x1 + 17$			IF_2	ID_2	EX_2	MA_2	WB_2			
(I_3)				IF_3	ID_3	EX_3	MA_3	WB_3		
(I_4)					IF_4	ID_4	EX_4	MA_4	WB_4	
(I_5)						IF_5	ID_5	EX_5	MA_5	WB_5

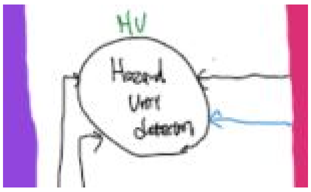
3. Forwarding Unit

Our forwarding unit is as follows:



T See section 2, Data Hazards, for more details.

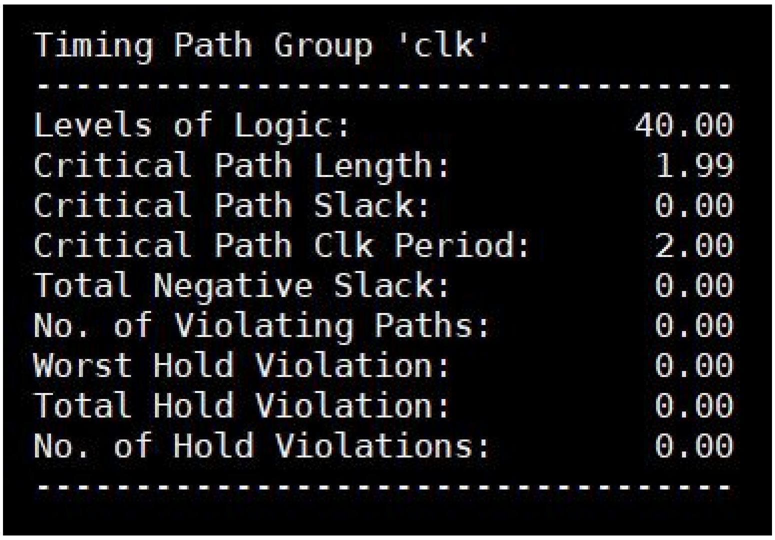
4. Hazard Detection Unit



When data hazard happens, the hazard detection unit figures it out and send a stall signal to wait until an operation is done.

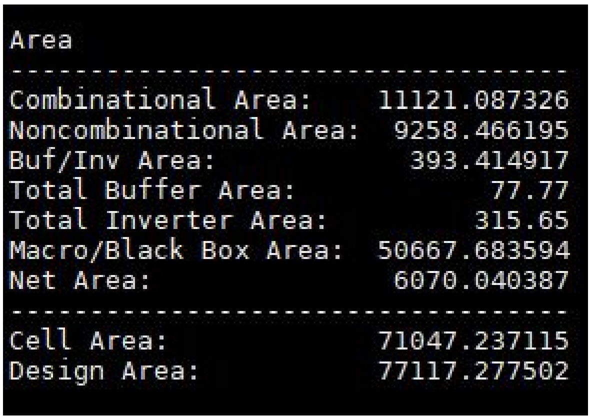
Synthesis

Critical Path Length / Critical Path Slack



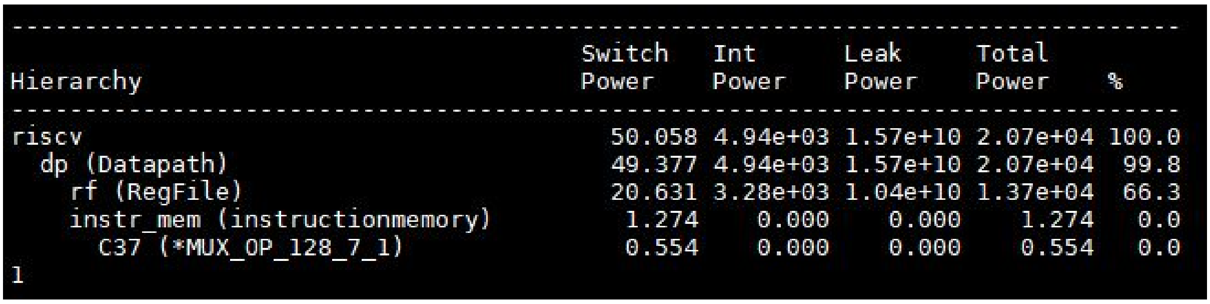
```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              40.00
Critical Path Length:          1.99
Critical Path Slack:           0.00
Critical Path Clk Period:      2.00
Total Negative Slack:          0.00
No. of Violating Paths:        0.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```

Area



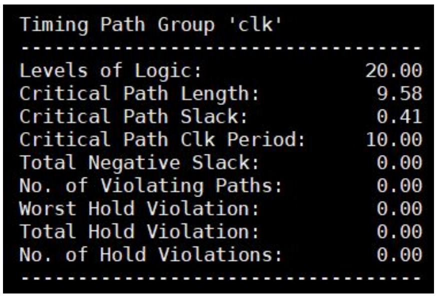
```
Area
-----------------------------------
Combinational Area:     11121.087326
Noncombinational Area:  9258.466195
Buf/Inv Area:            393.414917
Total Buffer Area:           77.77
Total Inverter Area:        315.65
Macro/Black Box Area:   50667.683594
Net Area:               6070.040387
-----------------------------------
Cell Area:              71047.237115
Design Area:            77117.277502
```

Power

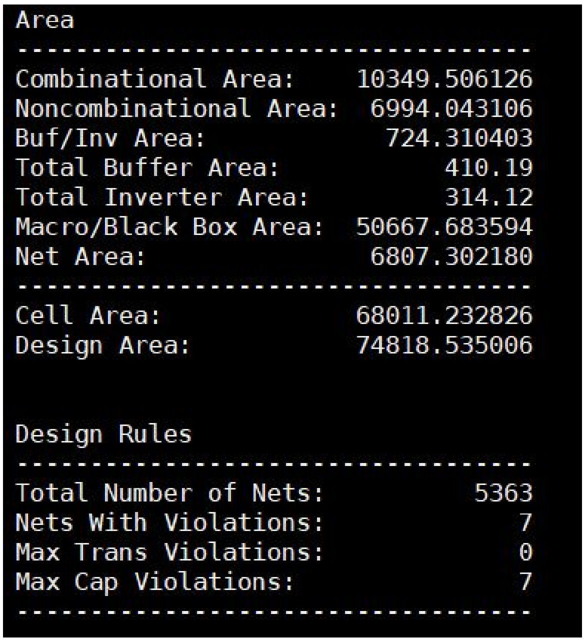


```
--------------------------------------------------------------------------------
                                       Switch   Int      Leak     Total
Hierarchy                              Power    Power    Power    Power    %
--------------------------------------------------------------------------------
riscv                                    50.058 4.94e+03 1.57e+10 2.07e+04 100.0
  dp (Datapath)                          49.377 4.94e+03 1.57e+10 2.07e+04  99.8
    rf (RegFile)                         20.631 3.28e+03 1.04e+10 1.37e+04  66.3
    instr_mem (instructionmemory)         1.274    0.000    0.000    1.274   0.0
      C37 (*MUX_OP_128_7_1)               0.554    0.000    0.000    0.554   0.0
1
```

Comparison with old synthesis

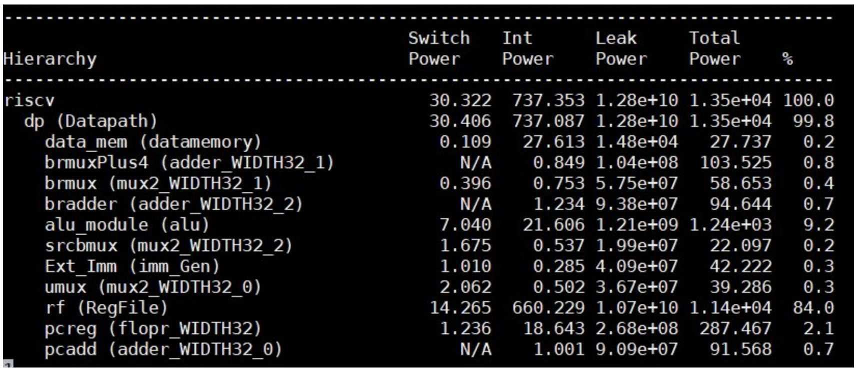


```
Timing Path Group 'clk'
-----------------------------------
Levels of Logic:              20.00
Critical Path Length:          9.58
Critical Path Slack:           0.41
Critical Path Clk Period:     10.00
Total Negative Slack:          0.00
No. of Violating Paths:        0.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------
```



```
Area
----------------------------------
Combinational Area:     10349.506126
Noncombinational Area:   6994.043106
Buf/Inv Area:             724.310403
Total Buffer Area:            410.19
Total Inverter Area:          314.12
Macro/Black Box Area:   50667.683594
Net Area:                6807.302180
----------------------------------
Cell Area:              68011.232826
Design Area:            74818.535006


Design Rules
----------------------------------
Total Number of Nets:           5363
Nets With Violations:              7
Max Trans Violations:              0
Max Cap Violations:                7
----------------------------------
```



```
--------------------------------------------------------------------------------
                                       Switch   Int      Leak     Total
Hierarchy                              Power    Power    Power    Power    %
--------------------------------------------------------------------------------
riscv                                    30.322  737.353 1.28e+10 1.35e+04 100.0
  dp (Datapath)                          30.406  737.087 1.28e+10 1.35e+04  99.8
    data_mem (datamemory)                 0.109   27.613 1.48e+04   27.737   0.2
    brmuxPlus4 (adder_WIDTH32_1)            N/A    0.849 1.04e+08  103.525   0.8
    brmux (mux2_WIDTH32_1)                0.396    0.753 5.75e+07   58.653   0.4
    bradder (adder_WIDTH32_2)               N/A    1.234 9.38e+07   94.644   0.7
    alu_module (alu)                      7.040   21.606 1.21e+09 1.24e+03   9.2
    srcbmux (mux2_WIDTH32_2)              1.675    0.537 1.99e+07   22.097   0.2
    Ext_Imm (imm_Gen)                     1.010    0.285 4.09e+07   42.222   0.3
    umux (mux2_WIDTH32_0)                 2.062    0.502 3.67e+07   39.286   0.3
    rf (RegFile)                         14.265  660.229 1.07e+10 1.14e+04  84.0
    pcreg (flopr_WIDTH32)                 1.236   18.643 2.68e+08  287.467   2.1
    pcadd (adder_WIDTH32_0)                 N/A    1.001 9.09e+07   91.568   0.7
```

In the previous version of the single cycle RISC-V model, it cannot run it in a small clock period. However, in the pipelined Risc_V, it can simulate in the clock period 2 and critical path length 1.99. From this information, the new pipelined processor works effectively better than the single cycle processor.