

Project 2: Pipelined CPU + L1 Data Cache

TA: 蔡承佑

r07922182@ntu.edu.tw

Project 1 Demo

- 4 set of test data (basic, forwarding, stall, flush)
- We will download your files and test them in our laptop
- timeslot sheet: <https://reurl.cc/R1NZez>



Announcement

- 1~3 persons in a group. Please check your group on NTU COOL
- **Deadline: 1/5 (Tue.) 23:59**
- Demo:
 - Time slot: TBD
 - Execute your program before TA and answer a few questions
 - All members in the group should attend

Specification

Use Verilog to model pipeline CPU with

- Off-chip Data Memory
 - Size: 16KB
 - Data width: 32 Bytes
 - Memory access latency: 10 cycles (send an ack when finish access)
- L1 Data Cache
 - Size: 1KB
 - Associative: 2-way
 - Replacement policy: LRU
 - Cache line size: 32 Bytes
 - Write hit policy: write back
 - Write miss policy: write allocate
 - offset: 5 bits, index: 4 bits, tag: 23 bits

testbench.v

- (optional) Initialize registers in all modules
- **Connecting CPU and off-chip Data_Memory**
- Load instruction.txt into instruction memory
- Create clock signal
- Dump Register files & Data memories in each cycle
- Print result to output.txt and cache.txt

Output Files

- Print result to **output.txt**
 - Output cache status when memory access occurs
 - Criteria: we will check **the final state** is correct or not (The cycle count does not matter)
- Print result to **cache.txt**
 - Record cache hit or cache miss for each cache access
 - Criteria: we will check **the order of hit and miss accesses** is identical to the correct answer (The cycle count does not matter)
- **DO NOT CHANGE THE OUTPUT FORMAT**

Grading Policy

- (80%) Programming
 - You will get 0 point if your code cannot be compiled
 - Grading at demo. You have to answer several questions about how you implement at demo. You may get 0 point on this part if you cannot clearly answer the questions (regarded as plagiarism)
- (20%) Report
 - Members & Team work (work division)
 - 務必寫組員分工比例
 - Implementation of modules
 - Cache controller in detail
 - You can draw a picture to explain if you want
 - Difficulties encountered and solutions of this projects
- Late punishment: 10 points deduction per day

Evaluation Criteria

```
4401 Flush Cache!
4402
4403 cycle = 200, Start = 1
4404 PC = 604
4405 Registers
4406 x0 = 00000000, x8 = eeef0021, x16 = 00000000, x24 = 00000000
4407 x1 = 00000000, x9 = ccccdfff, x17 = 22222222, x25 = 00000000
4408 x2 = 00000000, x10 = aaaabddd, x18 = 00000000, x26 = 00000000
4409 x3 = 00000000, x11 = 888899bb, x19 = 22222222, x27 = 00000000
4410 x4 = 00000000, x12 = 66667799, x20 = aaaaaaaa, x28 = 00000000
4411 x5 = 00000000, x13 = 44445577, x21 = 88888888, x29 = 00001000
4412 x6 = 00000000, x14 = 22223355, x22 = aaaaaaaa, x30 = 00000100
4413 x7 = 00000000, x15 = 00001133, x23 = 88888888, x31 = aaaaaaaa
4414 Data Memory: 0x0000 = 0000112222233444445566666778888899aaaaabccccccddeeeef0010
4415 Data Memory: 0x0020 = 88888888aaaaaaaa88888888aaaaaaaa2222222000000022222220000000
4416 Data Memory: 0x0040 = 0000000000000000000000000000000000000000000000000000000000000000
4417 Data Memory: 0x0200 = 0000000000000000000000000000000000000000000000000000000000000000
4418 Data Memory: 0x0220 = 0000000000000000000000000000000000000000000000000000000000000000
4419 Data Memory: 0x0240 = 0000000000000000000000000000000000000000000000000000000000000000
4420 Data Memory: 0x0400 = 000011332223355444557766667799888899bbbaaaabddccccddffeeef0021
4421 Data Memory: 0x0420 = 88888888aaaaaaaa88888888aaaaaaaa222222200000002222222000000000
4422 Data Memory: 0x0440 = 0000000000000000000000000000000000000000000000000000000000000000
```

```
4401 Flush Cache!
4402
4403 cycle = 200, Start = 1
4404 PC = 524
4405 Registers
4406 x0 = 00000000, x8 = eeef0021, x16 = 00000000, x24 = 00000000
4407 x1 = 00000000, x9 = ccccdfff, x17 = 22222222, x25 = 00000000
4408 x2 = 00000000, x10 = aaaabddd, x18 = 00000000, x26 = 00000000
4409 x3 = 00000000, x11 = 888899bb, x19 = 22222222, x27 = 00000000
4410 x4 = 00000000, x12 = 66667799, x20 = aaaaaaaa, x28 = 00000000
4411 x5 = 00000000, x13 = 44445577, x21 = 88888888, x29 = 00001000
4412 x6 = 00000000, x14 = 22223355, x22 = aaaaaaaa, x30 = 00000100
4413 x7 = 00000000, x15 = 00001133, x23 = 88888888, x31 = aaaaaaaa
4414 Data Memory: 0x0000 = 0000112222233444445566666778888899aaaaabccccccddeeeef0010
4415 Data Memory: 0x0020 = 88888888aaaaaaaa88888888aaaaaaaa2222222000000022222220000000
4416 Data Memory: 0x0040 = 0000000000000000000000000000000000000000000000000000000000000000
4417 Data Memory: 0x0200 = 0000000000000000000000000000000000000000000000000000000000000000
4418 Data Memory: 0x0220 = 0000000000000000000000000000000000000000000000000000000000000000
4419 Data Memory: 0x0240 = 0000000000000000000000000000000000000000000000000000000000000000
4420 Data Memory: 0x0400 = 000011332223355444557766667799888899bbbaaaabddccccddffeeef0021
4421 Data Memory: 0x0420 = 88888888aaaaaaaa88888888aaaaaaaa222222200000002222222000000000
4422 Data Memory: 0x0440 = 0000000000000000000000000000000000000000000000000000000000000000
```

For output.txt, we will only check the values of registers and data memory at the last cycle. You don't have to be exactly the same as reference output at every cycle.

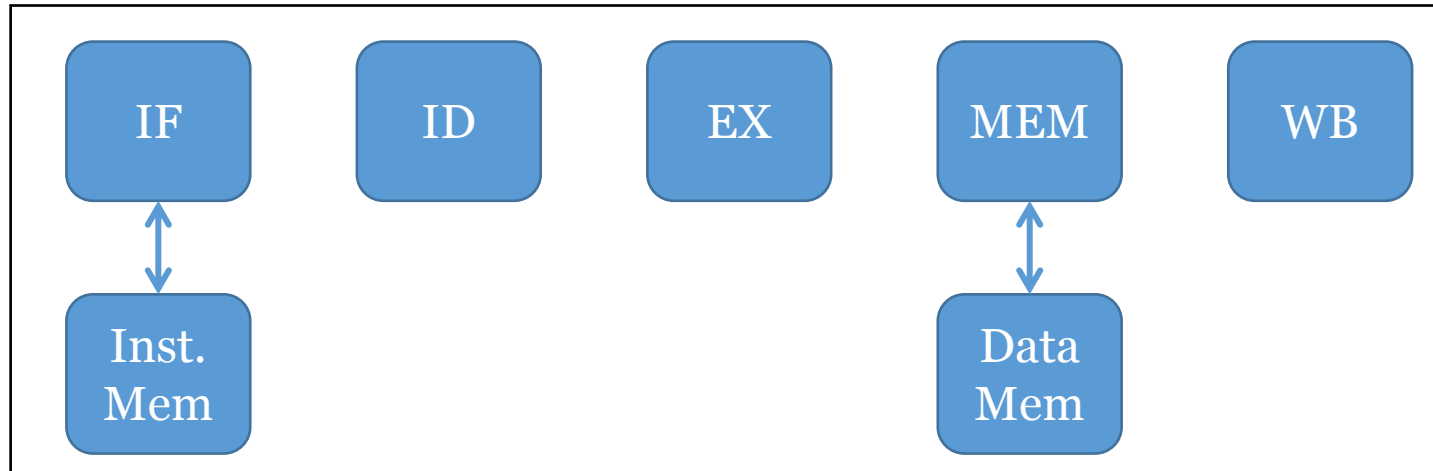
Evaluation Criteria (cont.)

1	Cycle:	3,	Read Miss	, Address: 00000000,	Read Data : xxxxxxxx
2	Cycle:	17,	Read Hit	, Address: 00000004,	Read Data : ccccdddd
3	Cycle:	18,	Read Hit	, Address: 00000008,	Read Data : aaaabbbb
4	Cycle:	19,	Read Hit	, Address: 0000000c,	Read Data : 88889999
5	Cycle:	20,	Read Hit	, Address: 00000010,	Read Data : 66667777
6	Cycle:	21,	Read Hit	, Address: 00000014,	Read Data : 44445555
7	Cycle:	22,	Read Hit	, Address: 00000018,	Read Data : 22223333
8	Cycle:	23,	Read Hit	, Address: 0000001c,	Read Data : 00001111
9	Cycle:	33,	Write Hit	, Address: 00000000,	Write Data: eeef0010
10	Cycle:	34,	Write Hit	, Address: 00000004,	Write Data: ccccddee
11	Cycle:	35,	Write Hit	, Address: 00000008,	Write Data: aaaabbcc
12	Cycle:	36,	Write Hit	, Address: 0000000c,	Write Data: 888899aa
13	Cycle:	37,	Write Hit	, Address: 00000010,	Write Data: 66667788
14	Cycle:	38,	Write Hit	, Address: 00000014,	Write Data: 44445566
15	Cycle:	39,	Write Hit	, Address: 00000018,	Write Data: 22223344
16	Cycle:	40,	Write Hit	, Address: 0000001c,	Write Data: 00001122
17	Cycle:	41,	Read Miss	, Address: 00000020,	Read Data : eeeeffff
18	Cycle:	54,	Read Hit	, Address: 00000024,	Read Data : 33332222
19	Cycle:	55,	Read Hit	, Address: 00000028,	Read Data : 55554444
20	Cycle:	56,	Read Hit	, Address: 0000002c,	Read Data : 77776666
21	Cycle:	57,	Read Hit	, Address: 00000030,	Read Data : eeeeffff
22	Cycle:	59,	Read Hit	, Address: 00000034,	Read Data : ccccdddd
23	Cycle:	60,	Read Hit	, Address: 00000038,	Read Data : aaaabbbb
24	Cycle:	61,	Read Hit	, Address: 0000003c,	Read Data : 88889999
25	Cycle:	81,	Write Hit	, Address: 00000020,	Write Data: 00000000
26	Cycle:	82,	Write Hit	, Address: 00000024,	Write Data: 22222222
27	Cycle:	83,	Write Hit	, Address: 00000028,	Write Data: 00000000
28	Cycle:	84,	Write Hit	, Address: 0000002c,	Write Data: 22222222
29	Cycle:	85,	Write Hit	, Address: 00000030,	Write Data: aaaaaaaa
30	Cycle:	86,	Write Hit	, Address: 00000034,	Write Data: 88888888
31	Cycle:	87,	Write Hit	, Address: 00000038,	Write Data: aaaaaaaa
32	Cycle:	88,	Write Hit	, Address: 0000003c,	Write Data: 88888888
33	Cycle:	97,	Write Miss	, Address: 00000400,	Write Data: eeef0021
34	Cycle:	110,	Write Hit	, Address: 00000404,	Write Data: ccccdfff
35	Cycle:	111,	Write Hit	, Address: 00000408,	Write Data: aaaabddd
36	Cycle:	112,	Write Hit	, Address: 0000040c,	Write Data: 888899bb
37	Cycle:	113,	Write Hit	, Address: 00000410,	Write Data: 66667799
38	Cycle:	114,	Write Hit	, Address: 00000414,	Write Data: 44445577
39	Cycle:	115,	Write Hit	, Address: 00000418,	Write Data: 22223355

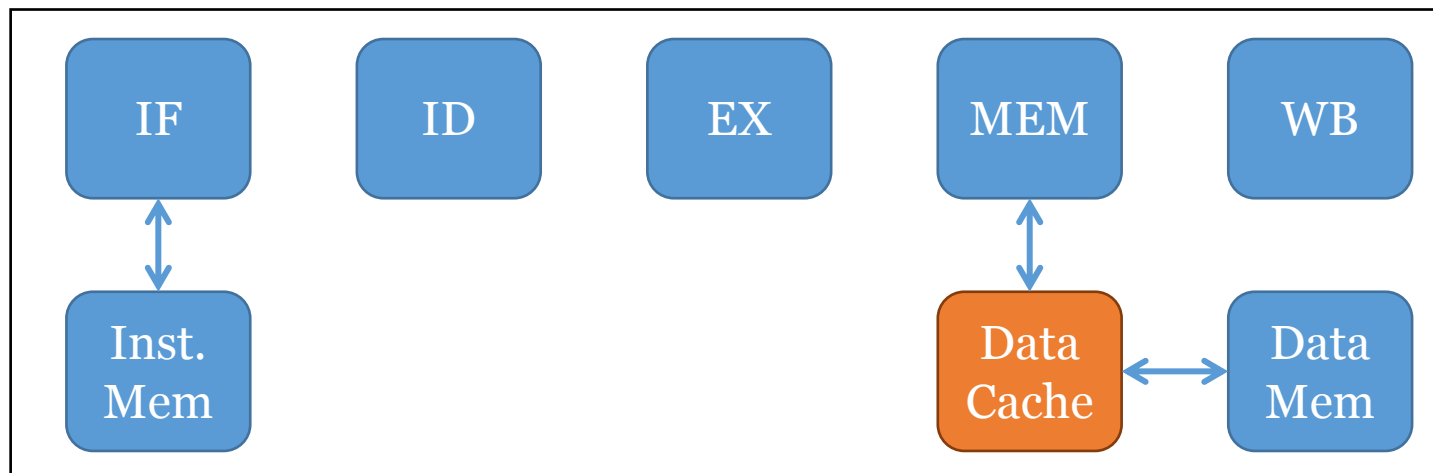
1	Cycle:	3,	Read Miss	, Address: 00000000,	Read Data : xxxxxxxx
2	Cycle:	16,	Read Hit	, Address: 00000004,	Read Data : ccccdddd
3	Cycle:	17,	Read Hit	, Address: 00000008,	Read Data : aaaabbbb
4	Cycle:	18,	Read Hit	, Address: 0000000c,	Read Data : 88889999
5	Cycle:	19,	Read Hit	, Address: 00000010,	Read Data : 66667777
6	Cycle:	20,	Read Hit	, Address: 00000014,	Read Data : 44445555
7	Cycle:	21,	Read Hit	, Address: 00000018,	Read Data : 22223333
8	Cycle:	22,	Read Hit	, Address: 0000001c,	Read Data : 00001111
9	Cycle:	31,	Write Hit	, Address: 00000000,	Write Data: eeef0010
10	Cycle:	32,	Write Hit	, Address: 00000004,	Write Data: ccccddee
11	Cycle:	33,	Write Hit	, Address: 00000008,	Write Data: aaaabbcc
12	Cycle:	34,	Write Hit	, Address: 0000000c,	Write Data: 888899aa
13	Cycle:	35,	Write Hit	, Address: 00000010,	Write Data: 66667788
14	Cycle:	36,	Write Hit	, Address: 00000014,	Write Data: 44445566
15	Cycle:	37,	Write Hit	, Address: 00000018,	Write Data: 22223344
16	Cycle:	38,	Write Hit	, Address: 0000001c,	Write Data: 00001122
17	Cycle:	39,	Read Miss	, Address: 00000020,	Read Data : eeeeffff
18	Cycle:	52,	Read Hit	, Address: 00000024,	Read Data : 33332222
19	Cycle:	53,	Read Hit	, Address: 00000028,	Read Data : 55554444
20	Cycle:	54,	Read Hit	, Address: 0000002c,	Read Data : 77776666
21	Cycle:	55,	Read Hit	, Address: 00000030,	Read Data : eeeeffff
22	Cycle:	57,	Read Hit	, Address: 00000034,	Read Data : ccccdddd
23	Cycle:	58,	Read Hit	, Address: 00000038,	Read Data : aaaabbbb
24	Cycle:	59,	Read Hit	, Address: 0000003c,	Read Data : 88889999
25	Cycle:	78,	Write Hit	, Address: 00000020,	Write Data: 00000000
26	Cycle:	79,	Write Hit	, Address: 00000024,	Write Data: 22222222
27	Cycle:	80,	Write Hit	, Address: 00000028,	Write Data: 00000000
28	Cycle:	81,	Write Hit	, Address: 0000002c,	Write Data: 22222222
29	Cycle:	82,	Write Hit	, Address: 00000030,	Write Data: aaaaaaaa
30	Cycle:	83,	Write Hit	, Address: 00000034,	Write Data: 88888888
31	Cycle:	84,	Write Hit	, Address: 00000038,	Write Data: aaaaaaaa
32	Cycle:	85,	Write Hit	, Address: 0000003c,	Write Data: 88888888
33	Cycle:	94,	Write Miss	, Address: 00000400,	Write Data: eeef0021
34	Cycle:	107,	Write Hit	, Address: 00000404,	Write Data: ccccdfff
35	Cycle:	108,	Write Hit	, Address: 00000408,	Write Data: aaaabddd
36	Cycle:	109,	Write Hit	, Address: 0000040c,	Write Data: 888899bb
37	Cycle:	110,	Write Hit	, Address: 00000410,	Write Data: 66667799
38	Cycle:	111,	Write Hit	, Address: 00000414,	Write Data: 44445577
39	Cycle:	112,	Write Hit	, Address: 00000418,	Write Data: 22223355

For cache.txt, we will only check number of hit/miss and their order.
cycle count doesn't matter.

Project 1 to Project 2



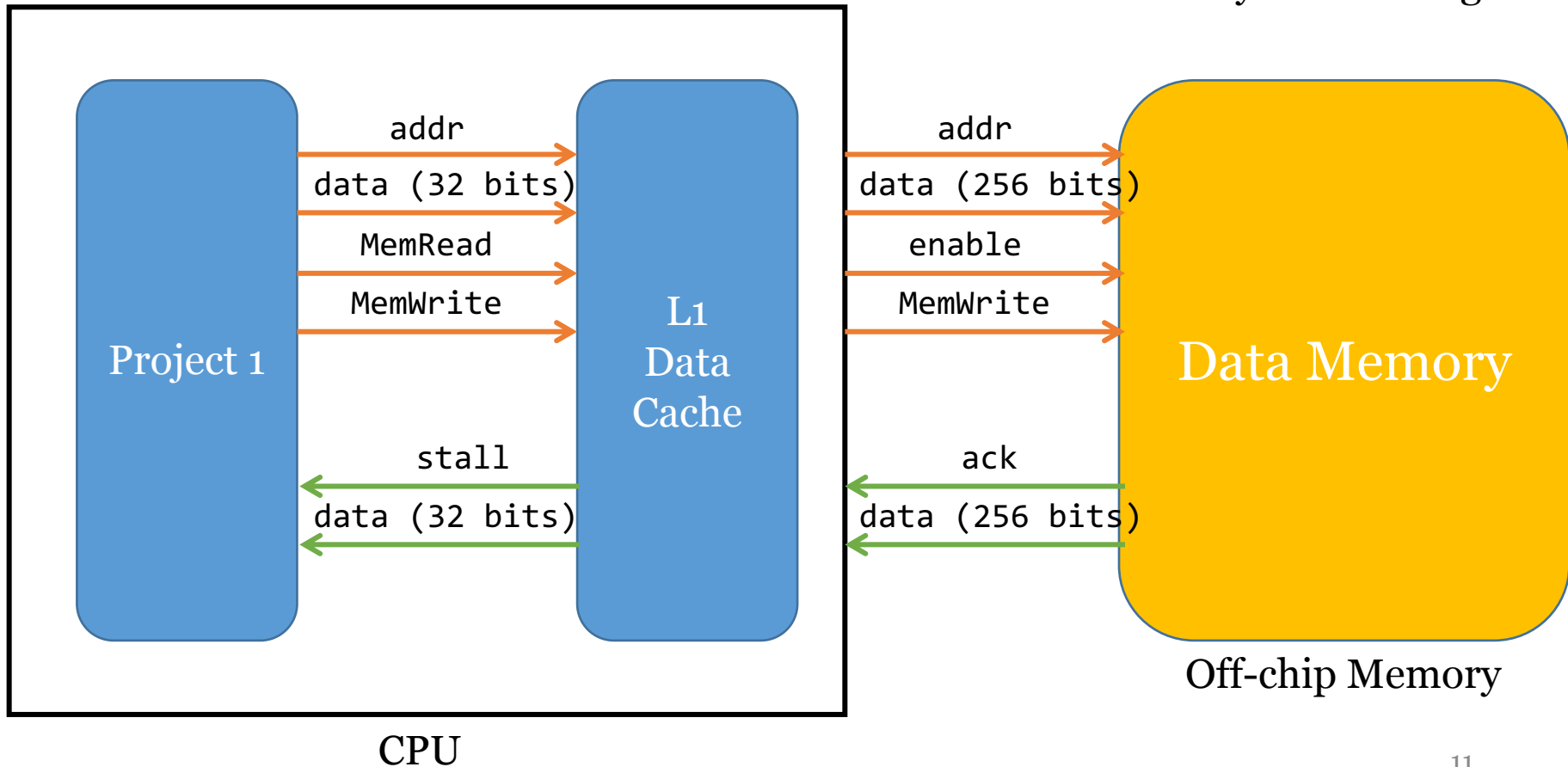
Project 1



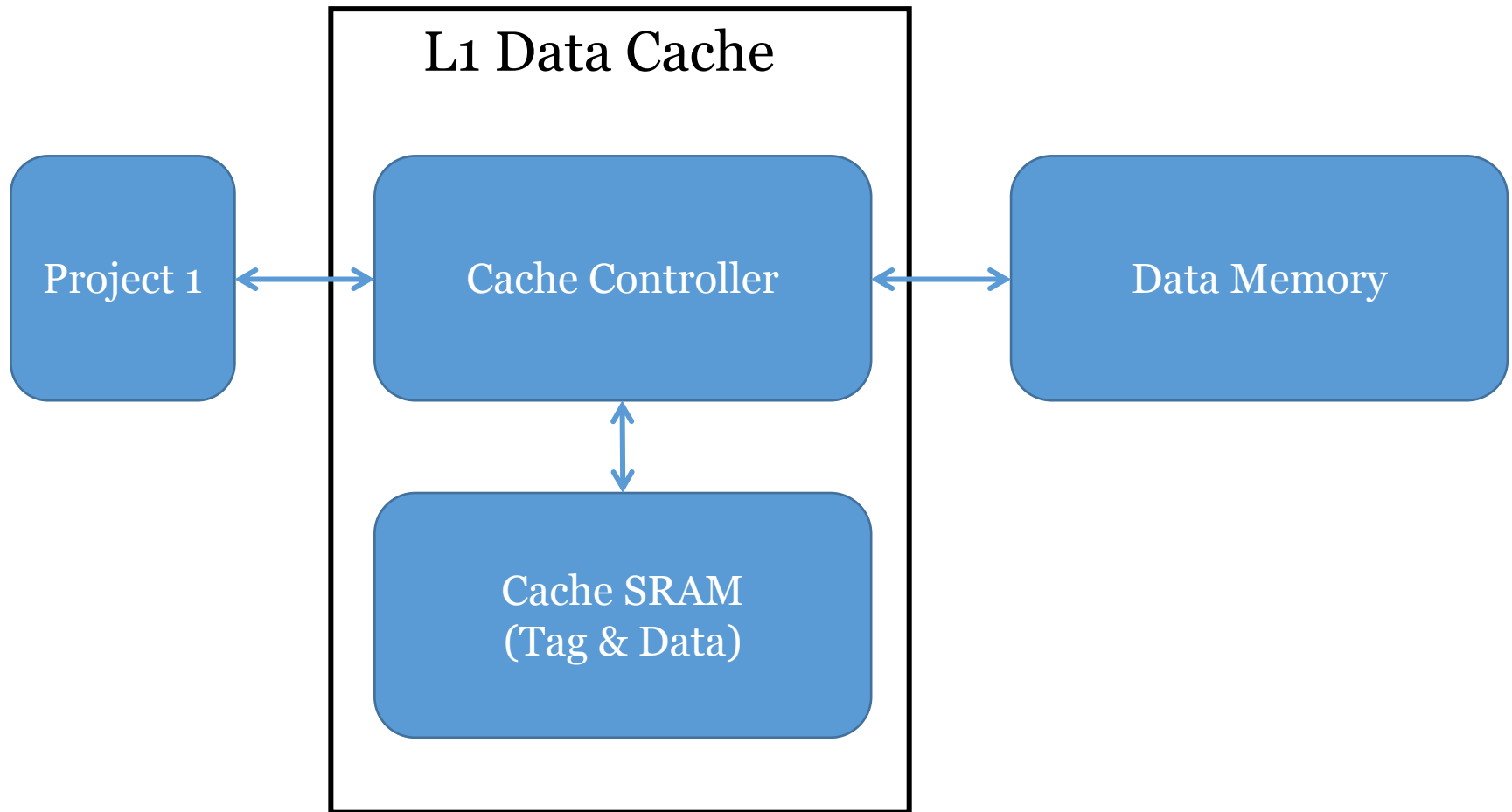
Project 2

System Block Diagram

enable: memory access enable
write: write data to memory
ack: memory acknowledge



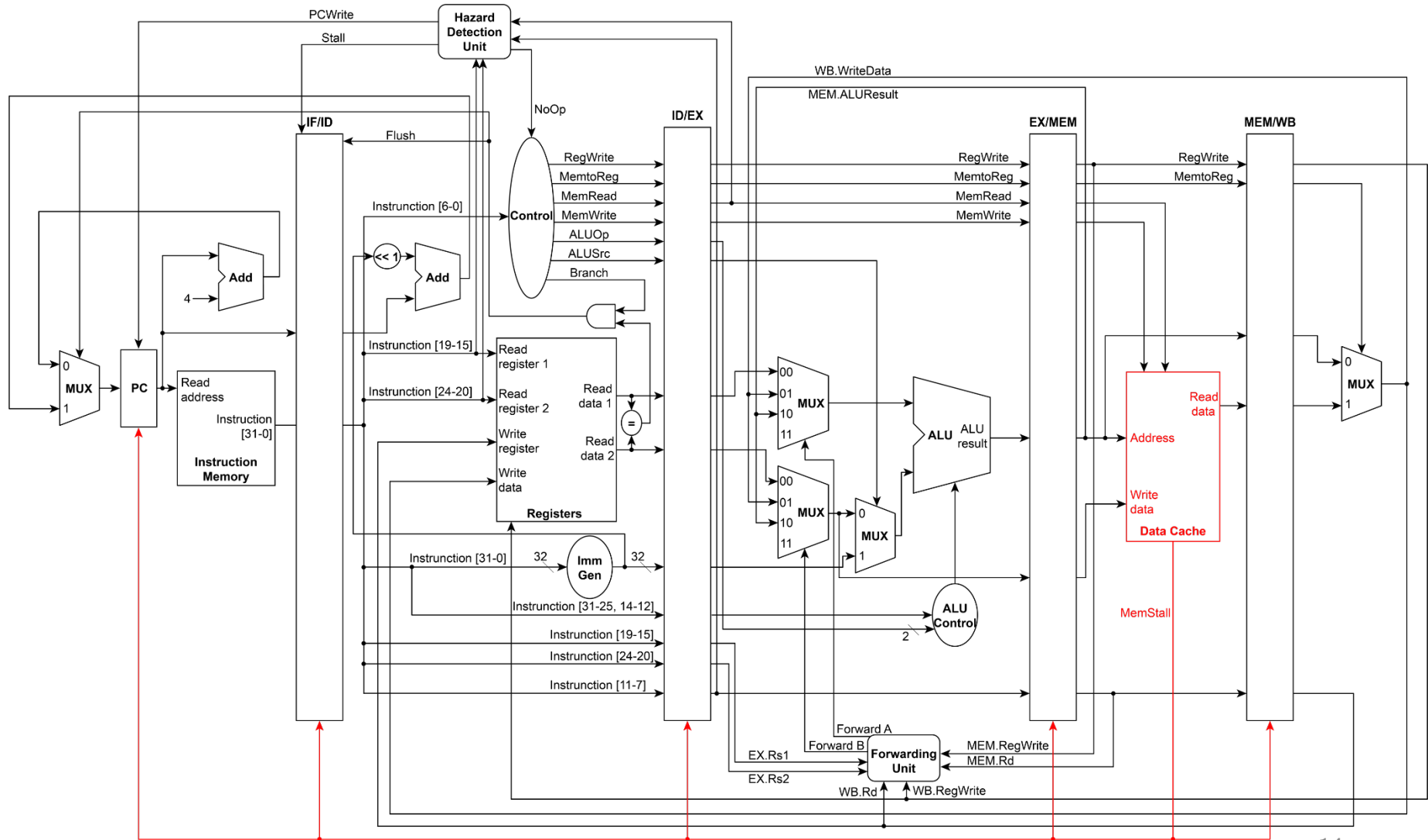
Example files



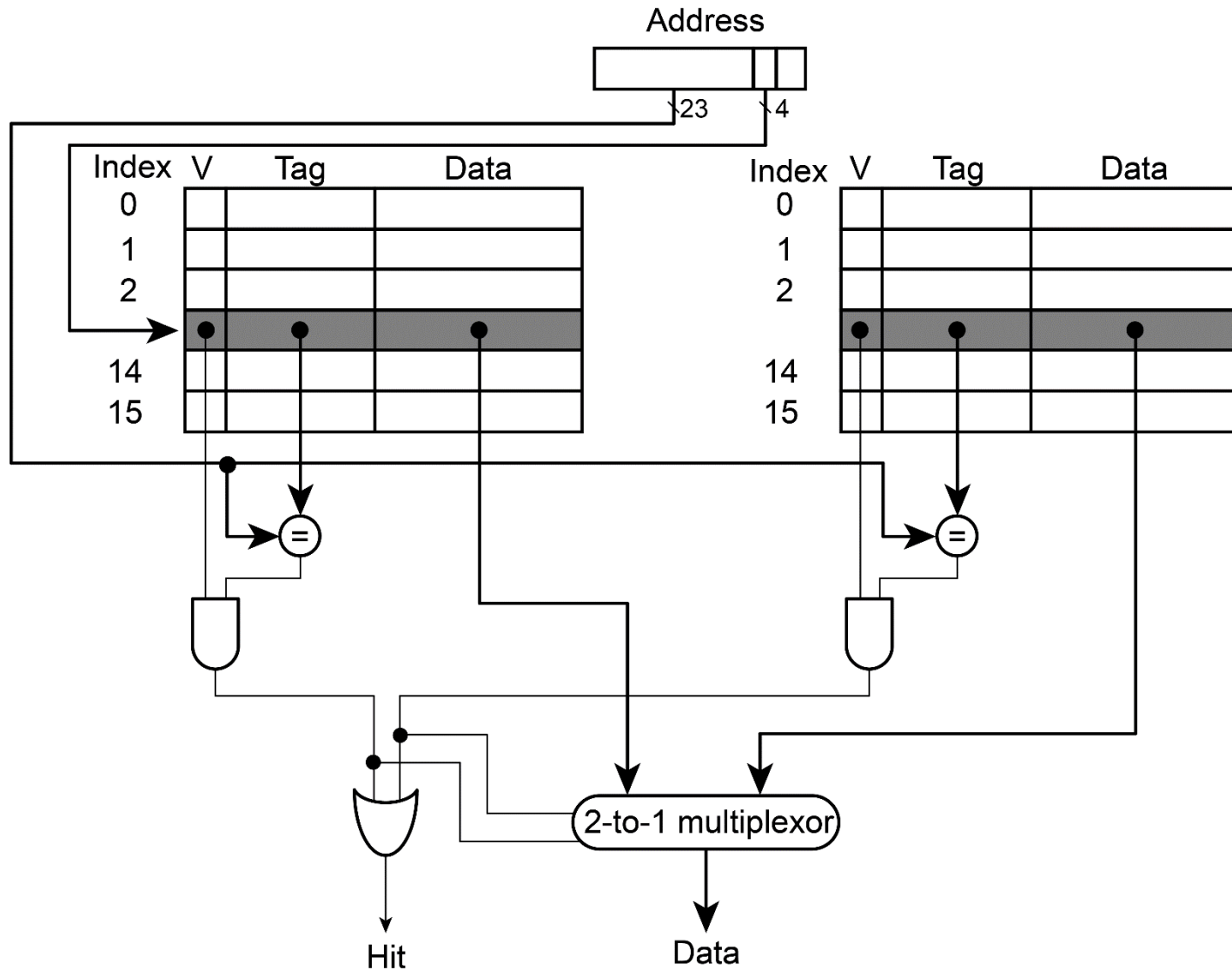
Example files

- `CPU.v`: connection between modules
 - `dcache_controller.v`: handles I/O requests
 - `dcache_sram.v`: Modify the data structure within it to support 2-way associative cache.
 - `Data_Memory.v`
 - `testbench.v`
 - `Instruction_Memory.v`
 - `PC.v`
 - `Registers.v`
- You can modify them as you want
But make sure you include them as submission

Datapath & Modules



2-way associative cache



Submission Rules

- teamXX_project2 (dir)
 - teamXX_project2/codes/*.v
 - teamXX_project2/teamXX_project2_report.pdf
- Pack the above **directory** into a zip file
 - When we unzip your file, the output should be a single directory named teamXX_project2
- You can fill in whatever you want in teamXX, but please use ASCII printable characters.

Submission Rules (cont.)

- In testbench.v, you must check the following settings before submission
 - Read instruction from `./instruction.txt`
 - Dump output to `./output.txt` and `./cache.txt`
- Your code can be compiled with the follow command
 - `$ iverilog -o CPU.out *.v`

Appendix: COOL comments

近期 NTU COOL 通知



NTU COOL <ntu-cool@cool.ntu.edu.tw>
To: 蔡承佑

Yesterday at 10:00

To protect your privacy, some pictures in this message were not downloaded. [Download pictures](#)

您已註冊接收來自您 NTU COOL 帳戶一些通知的每日報告。以下是 12月15日 的報告：

提交項目評論：[redacted]，Project 1 Building Pipelined CPU，計算機結構 Computer Architecture

[redacted] 剛剛對 Project 1 Building Pipelined CPU 的 [redacted] 的提交項目進行了新評論。

[按一下以檢視](#)

請注意：此郵件是系統自動傳送，請勿直接回覆！

Please mail us directly instead of using COOL comments to ask your questions

[更新您的通知設定](#)

Appendix: Email Etiquette Rules

From:	r07922182@ntu.edu.tw (蔡承佑)
To:	eclabta@gmail.com
Cc:	1. CC your teammates
Subject:	[Project 2] 2. Proper title with a tag

| 3. Clearly specify your question

4. Sign your name