Digital System Design Homework#4
Cache Unit Design
Report **B05901030 電機四 陳欽安**

(1) Direct mapped:

I. Cycle time: 10.0ns

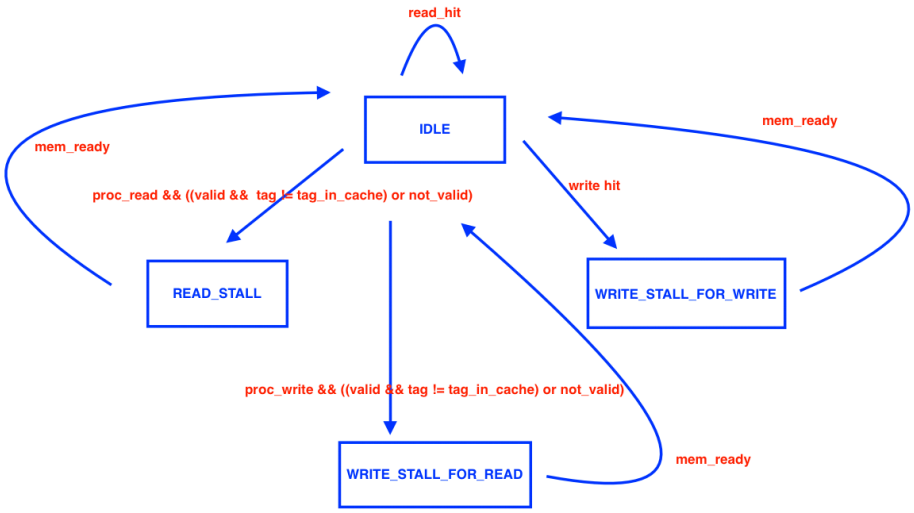
```
`timescale 1 ns/10 ps
`define CYCLE       10.0               // Modify cycle time here
`define SDFFILE    "./cache_dm_syn.sdf"   // Modify your sdf file name
==== CONGRATULATIONS! Pass cache read-write-read test. ====

Finished all operations at:               158805 ns
Exit testbench simulation at:             158905 ns


Simulation complete via $finish(1) at time 158905 NS + 0
```

II. General specification of the cache unit:
   a) Numbers of words
      32-word cache consists of 8 blocks, each containing 4 words.
   b) Placement policy
      format: reg[153:0] cache[0:7]

153	152:128	127:96	95:64:	63:32	31:0
valid bit	tag	word3	word2	word1	word0

III. Read/ Write policy: Write- through (also update memory)

IV. Design architecture or the finite state machine of the cache unit



V. Performance

```
Stall cycle:          6400
Read miss  :           256
Write miss :          1024
read cnt:             1024
write cnt:            1024
```

miss rates of read = 256/1024 = 25%
miss rates of write = 1024/1024 = 100%
total cycle:
   read: 1024 + 256 x 5 = 2304
   write: 1024 + 1024 x 5 = 6144
   total: 2304+ 6144 = 8448
Stall cycle: 6400

(2) 2-way associative:

I. Cycle time: 10.0ns

```
`timescale 1 ns/10 ps
`define CYCLE       10.0              // Modify cycle time here
`define SDFFILE    "./cache_2way_syn.sdf"   // Modify your sdf file name
==== CONGRATULATIONS! Pass cache read-write-read test. ====

Finished all operations at:                120405 ns
Exit testbench simulation at:              120505 ns


Simulation complete via $finish(1) at time 120505 NS + 0
```

II. General specification of the cache unit:

a) Numbers of words

32-word cache consists of 4 sets, each set consists of 2 blocks, and each block contains 4 words.

b) Placement policy

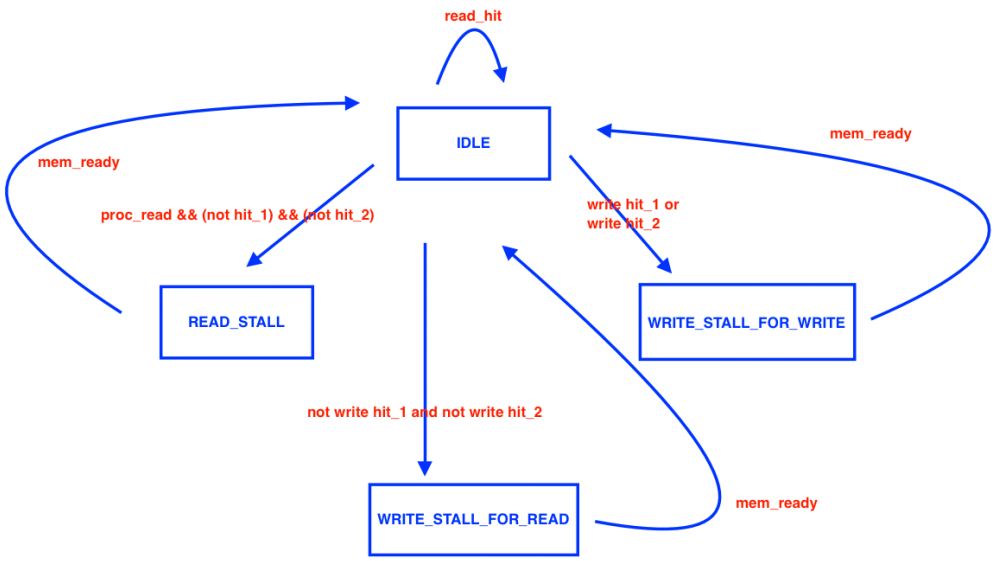
format: reg[311:0] cache[0:3]

311	310	309:284	283:156	155	154	153:128	127:0
valid_1	LRU_1	tag_1	word3~word0	valid_2	LRU_2	tag_2	word3~word0

where LRU records whether the block is recently used.

III. Read/ Write Policy: Write- through (also update memory)

IV. Design architecture or the finite state machine of the cache unit



V. Performance

```
Stall cycle:          2560
Read miss  :           256
read cnt:             1024
write cnt:            1024
```

Miss rate of read = 256/1024 = 25%
Miss rate of write = 256/1024 = 25%
Total cycle:
read: 1024 + 256 x 5 = 2304
write: 1024 + 256 x 5 = 2304
total: 2304 + 2304 = 4608
Stall cycle: 2560

(3) Compare the performance of the two architectures, and discuss the reasons for such results

因為2-way associativity 可以降低miss發生的機率，就是減少conflict的發生，
所以從performance可以看出 2-way associativity 的miss rate on write 以及總stall cycle都比direct map來得少。

BONUS: L2 cache

#readme:

test RTL:

ncverilog tb_cache.v bonus.v memory.v+access+r

test gate-level:

edit in tb_cache.v

- SDFFILE: “./bouns.sdf”
- CYCLE 12.0

```
`timescale 1 ns/10 ps
`define CYCLE       12.0              // Modify cycle time here
`define SDFFILE    "./bonus_syn.sdf"  // Modify your sdf file name
```

ncverilog tb_cache.v bonus_syn.v memory.v tsmc13.v +define+SDF +access+r

Direct mapped: 128-word cache consists of 32 blocks, each containing 4 words.

Pass RTL/ gate-level simulation:

```
[b05030@cad29 HW4]$ ncverilog tb_cache.v bonus.v memory.v tsmc13.v +access+r
==== CONGRATULATIONS! Pass cache read-write-read test. ====

Finished all operations at:                144486 ns
Exit testbench simulation at:              144606 ns


Simulation complete via $finish(1) at time 144606 NS + 0
./tb_cache.v:211         $finish;
```

```
[b05030@cad29 HW4]$ ncverilog tb_cache.v bonus_syn.v memory.v tsmc13.v +define+SDF +access+r
==== CONGRATULATIONS! Pass cache read-write-read test. ====

Finished all operations at:                144486 ns
Exit testbench simulation at:              144606 ns


Simulation complete via $finish(1) at time 144606 NS + 0
./tb_cache.v:211         $finish;
```

Performance compare with L1 directed map:

direct map only L1 cache :

```
Stall cycle:        6400
Read miss  :         256
Write miss :        1024

read cnt:         1024
write cnt:        1024
```

direct map with L1 and L2 cache:

```
Stall cycle:        2560
Read miss  :         256
Write miss :         256

read cnt:         1024
write cnt:        1024
```

-> write miss decrease extremely-> because L2 cache focus on low miss rate to avoid main memory access.