CA HW5 Report

B03901022 卓伯鴻

(a) General specification of the cache unit(such as numbers of words, replacement policy, and so on);

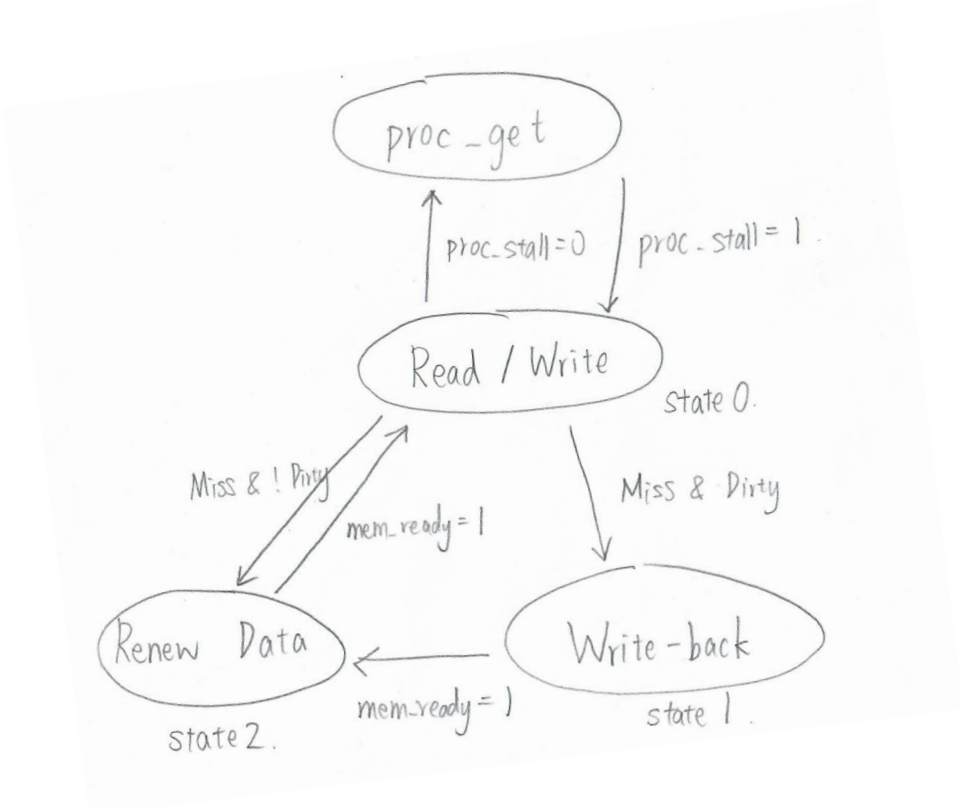
Ans:
1. numbers of words: There are 8 blocks in the cache, and each block has 4 words in it.
2. replacement policy: Since the cache is direct-mapped, there are no replacement policy. For each “proc_addr”, there are only one block pointed, which is decided by its own module of the [4:2] bit in “proc_addr”.
3. block design: For each block, there exists 1 bit of “valid bit” to indicate whether the block is stored with valid values from memory, 1 bit of “dirty bit” to indicate whether the block is changed since the value is retrieved from memory, 25 bits of “tag bits” to indicate the correctness of “proc_addr” when the block is pointed, and finally 128 bits of “word bits” to store 4 words in the block.

(b) Read/write policy (write-through or write-back);

Ans:
policy: I used “write-back” policy in the cache, which I only writes the value stored in cache back to memory when the block is pointed and is missed and the block have been changed(a.k.a. ‘dirty bit’ of the block is 1’b1).

(c) Design architecture or the finite state machine of the cache unit;

Ans:



The plot above is the perception of my finite state machine. The main processing state is state0: read/ write state. It determines whether to return data/ write data and go to state 3: proc_get state, to go to state 2: write-back state, or to go to state 3: renew data state. In write-back state, the cache write back the changed values in block to the memory. In renew-data state, the cache is filled with new data retrieved from the memory. In proc_addr state, the proc_stall is set to 1'b0, and new instruction was fetched in.

(d) Performance evaluation of your cache design, including the miss rates of(read/write operations, the execution cycles, the stalled cycles, and so on.)

Ans:
miss rate: Read operation has 25% (256/ 1024) miss rate, and write operation has 25%(256/ 1024) miss rate.
cycles: If the proc_addr hits, it last for 2 cycles. If the proc_addr misses, and the dirty bit is 1'b0, it last for 8 cycles(only go to state 2 and return with new values from the memory). If the proc_addr misses, and the dirty bit is 1'b1, it last for 13 cycles(go to write-back state and then go to renew-data state), which is the longest one among all three possibilities.

(e) Please annotate the clock cycle for passing post synthesis simulation, TA will run the simulation according to the clock cycle in report.

Ans:
I use cycle = 5 ns in synthesis process, and I find 8(ns) per cycle in test_bench is the smallest integer time per cycle can pass the lower bound of violation of time.

(f) Report from synthesis process:

```
Combinational area:      37650.028674
Noncombinational area:   47311.628597
Net Interconnect area:   763330.571808

Total cell area:         84961.657271
Total area:              848292.229079
```

```
clock CLK (rise edge)                      5.00       5.00
clock network delay (ideal)                0.50       5.50
clock uncertainty                         -0.10       5.40
block_reg[2][3][30]/CK (DFFRX1)            0.00       5.40 r
library setup time                        -0.16       5.24
data required time                                    5.24
-----------------------------------------------------------
data required time                                    5.24
data arrival time                                    -5.23
-----------------------------------------------------------
slack (MET)                                           0.01
```