Pipelined CPU with L1-Cache using Verilog

-- Computer Architecture Project 2

■ Team member

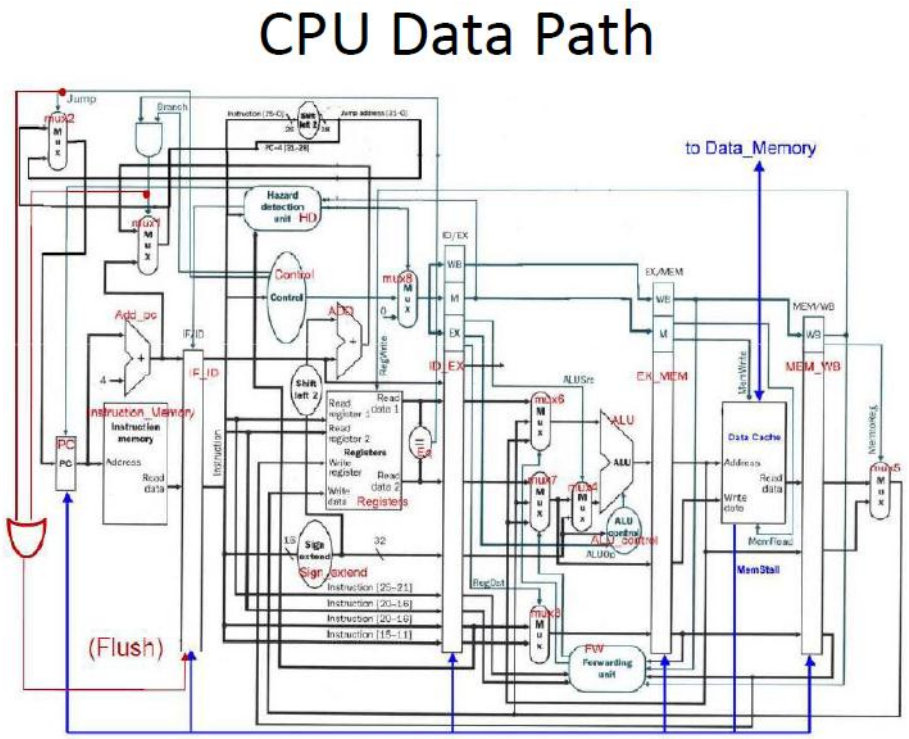
趙愷 T01902201;T01902201@ntu.edu.tw

■ Implement of CPU

Basically, the CPU is made of four components.

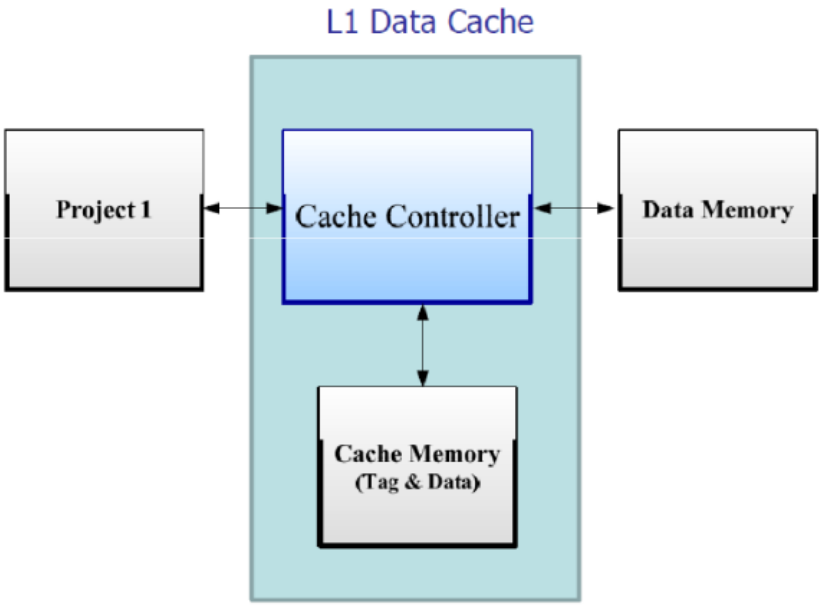
1. **Control logic :** Control for pc,
   Selector and other unit,
   Hazard detection unit,
   Forward unit.
2. **Memory:** PC, Instruction_Memory and L1-Cache.
3. **Registers:** There are 32 registers in mips architecture.
   And data in each stage are also store in registers called 'ID_EX'…
4. **ALU:** Use for compute

The picture below shows Data Path and Module for this CPU in detail.

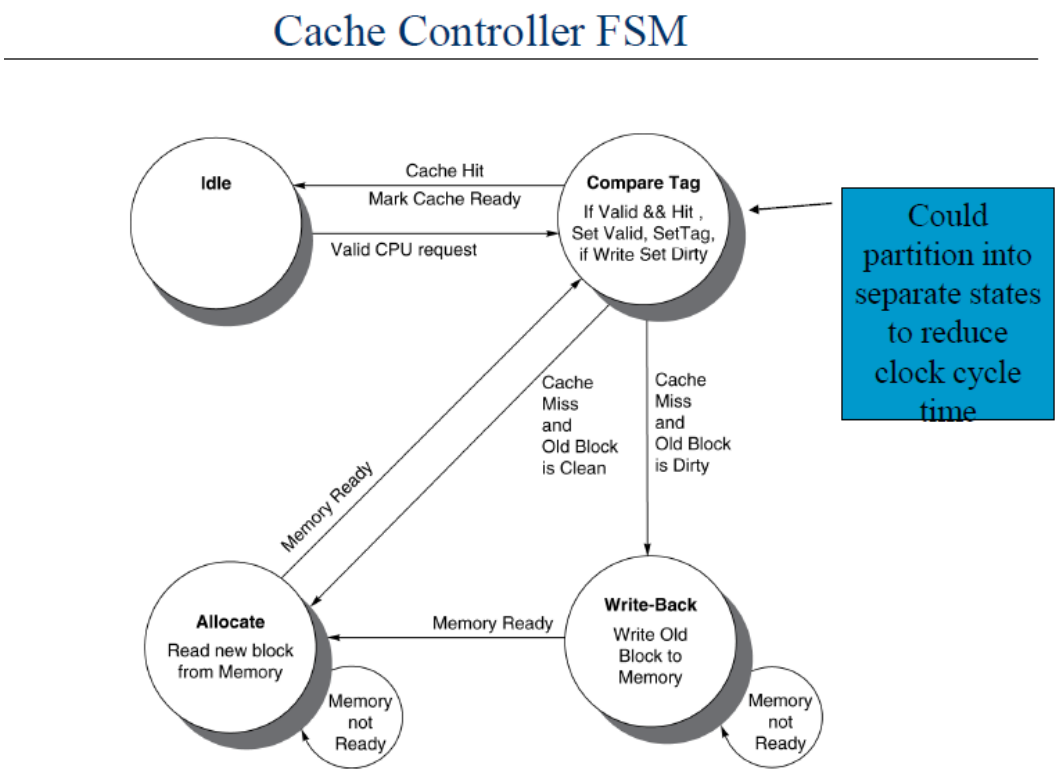


■ Implement of Cache

The cache module is made up of three components:
Cache Controller, Cache Memory of Tag, Cache Memory of Data.
And their **relationship** is as below:



Cache Controller use FSM to conduct the cache function:



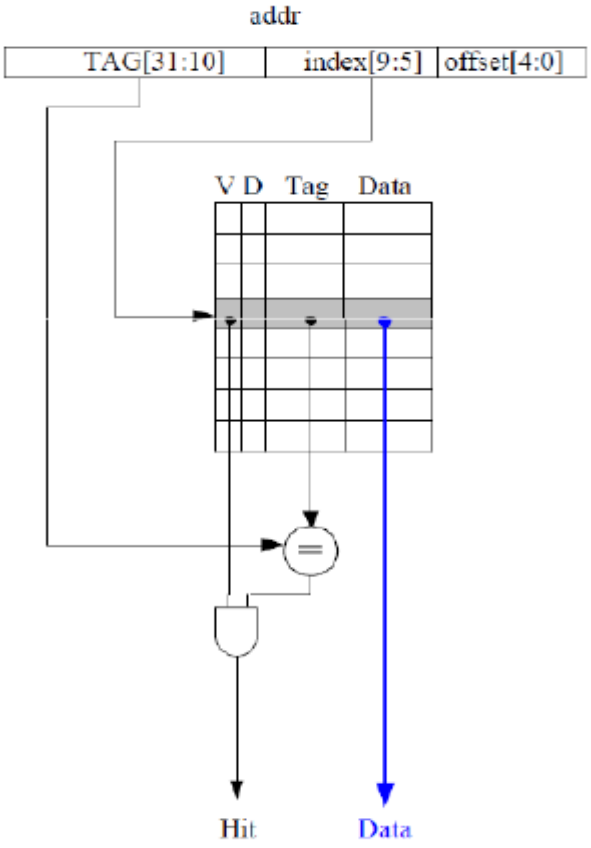
The procedure of a cache hit/miss:

The picture in right side shows how we compute whether the address is hit or not:

If tag is equal and the valid bit is set, the address is hit. And we get the data from cache or write data to cache(write back).

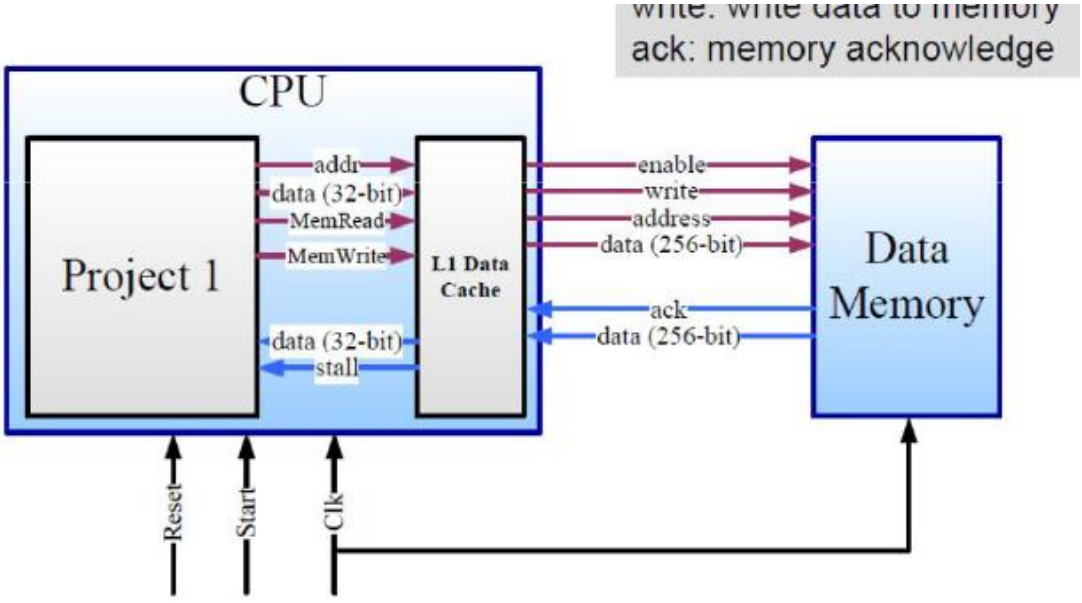
If miss, first we write the old data to memory if its dirty bit is set. Then wait memory prepare the data and send ack.

If this is a read miss, it`s done. If this is a write miss, now it turn to a write hit, next cycle the write data will be write into cache.



■ Connection of CPU with L1-Cache and Data Memory

Finally, we get the CPU with L1-Data Cache, the picture below shows the connection among major components.



■ Problems and solution

Problem 1.

In order to select 32bit data from the 256bit r_hit_data according to p1_offset, I use the Variable Part Select method which is added inVerilog 2001:

My Code

```
// read data :    256-bit to 32-bit
always@(p1_offset or r_hit_data) begin
      p1_data =hit ? r_hit_data[(p1_offset>>2)*32+31 -: 32] : 32'b0;
end
```

Variable Part Select (added in Verilog-2001)

vector_name[starting_bit_number +: part_select_width]

vector_name[starting_bit_number -: part_select_width]

- Variable part selects can vary the starting point of the part select, but the width of the part select must be a literal number, a constant or a call to a constant function. Variable part selects were added in Verilog-2001.
- +: indicates the part select increases from the starting point.
- -: indicates the part select decreases from the starting point.

---from Verilog® HDL Quick Reference Guide based on the Verilog-2001 standard (IEEE Std 1364-2001),
See http://www.sutherland-hdl.com/online_verilog_ref_guide/verilog_2001_ref_guide.pdf