

1. Instruction memory, Registers, ALU, 因為它們在執行sd指令的 cycle中, 只會根據input得到output, 就好像是 (act as) combinational circuits一樣。 (5%)
2. 在原本的design中, controller的input只有7個bit的opcode, ALU control的input則是1-bit的funct7以及3-bit的funct3, 跟一次使用17-bit造出單一controller的design比較起來, 電路會比較小。換句話說, 也就是製造一個input為17-bit的controller會比製造一對input分別為7 bits及4 bits的controllers來得大許多, 效率也就差了。 (10%)
3. (a)為了處理beq的target address為 $PC + imm|0$ (7%)
(b)Prevent update of PC and IF/ID register + Force control signals in ID/EX register to 0 (8%)
(c-1)add這個instruction以及control signal
(c-2)EX用的X4的值為對的, 因為是從ld forward過來的)
MEM的ADD所用的X4的值為錯的
(c-3)yes (4%, 4%, 4%)
(d)complete the instructions previous than LD->
flush LD and subsequent instructions->
set SEPC and SCAUSE register values->
Transfer control to handler->
After handler executes, returns to the LD instruction (2% for each)
(e) (8%)
4. (a) tag = 48bit, index = 13bit (4%)
(b) 因為只有一個地方能寫入, 就直接蓋過去, 所以不用check (因為不管是write hit or miss, write-through的direct-mapped cache都是直接寫到cache上(只有一個可能的位置), 接著更新memory, 所以檢查是不是write hit沒有意義) (4%)
(c)要先check replace的地方有無東西:
有東西但不是dirty就可以直接放
有東西但是dirty就要先寫回memory才能放
若沒東西則可以直接放 (4%)
(d)加大block (4%)
5. (a)0x0000F9C0DE4B5 (6%)
(b)在此題為NO, 因為page大小<64KB, (3%)
(c)加大 associativity (3%)
6. (a) $0.9 + 0.06 * 70 = 5.1$ (6%)

because clock rate = 1ns 所以答案為5.1cycles (6%)

$$(b) 0.9 + 0.06 * 5.62 + 0.06 * 0.05 * 70 = 1.4472$$

答案為1.4472cycles