COMPSCI 2GA3 Fall, 2021

Final Exam, Dec. 20th 2021

1. Question (10 marks)

Consider two different processors P1 and P2 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 4.0 GHz clock rate and has a CPI of 2.

a) (4 marks) Which processor has the highest performance expressed in instructions per second?

IPS_1 = clock rate$_1$ / CPI$_1$ = 3GHz / 1.5 = 2×10^9
IPS_2 = clock rate$_1$ / CPI$_2$ = 4GHz / 2 = 2×10^9
Processors have the same performance rate

b) (4 marks) If each processor execute a program in 7 seconds, find the number of cycles and the number of instructions.

$instructions_1$ = IPS$_1$ × CPU timer$_1$ = $2 \times 10^9 \times 7 = 1.4 \times 10^{10}$
clock cycles$_1$ = CPU timer$_1$ × clock rate$_1$ = 7 × 3GHz = $7 \times 3 \times 10^9 = 2.1 \times 10^{10}$

$instructions_2$ = IPS$_2$ × CPU timer$_2$ = $2 \times 10^9 \times 7 = 1.4 \times 10^{10}$
clock cycles$_2$ = CPU timer$_2$ × clock rate$_2$ = 7 × 4GHz = $7 \times 4 \times 10^9 = 2.8 \times 10^{10}$

c) (2 marks) We are trying to reduce the execution time by 20%, but this leads to an increase of 25% in the CPI. What clock rate should we have to get this time reduction?

Execution time$_{new}$ = 0.80 Execution time$_{old}$

$$\frac{instructions_{new} \times CPI_{new}}{clock\ rate_{new}} = 0.80\ \frac{instructions_{old} \times CPI_{old}}{clock\ rate_{old}}$$

Notice that

$$instructions_{new} = instructions_{old}$$

$$\frac{CPI_{new}}{clock\ rate_{new}} = 0.8\ \frac{CPI_{old}}{clock\ rate_{old}}$$

now we use that

$$CPI_{new} = 1.25\ CPI_{old}$$

to get

$$\frac{1.25}{clock\ rate_{new}} = \frac{0.8}{clock\ rate_{old}}$$

And finally rearranging that

$$clock\ rate_{new} = \frac{1.25}{0.8} \times clock\ rate_{old} = 1.5625 \times clock\ rate_{old}$$

Therefore clock rate must increase by 56.25%.

2. Question (10 marks)

Translate function `f` into RISC-V assembly language. The code for function `f` is as follows:

```
f(long g, long i, long A[], long B[])
{

return  g + A[B[i] + A[1]];

}
```

Assume arguments `g,i` are in registers `x11,x12`. Base addresses of the arrays A and B are in registers `x6` and `x7` respectively. Return value should be in `x10`. Please write comment at each line of the code. We assume that, i ¡ size fo array B, and 0¡= B[i] - A[1] B[i] - A[1] ¡= size of array A.
Hint: If you need temporal registers you should save them on the stack.

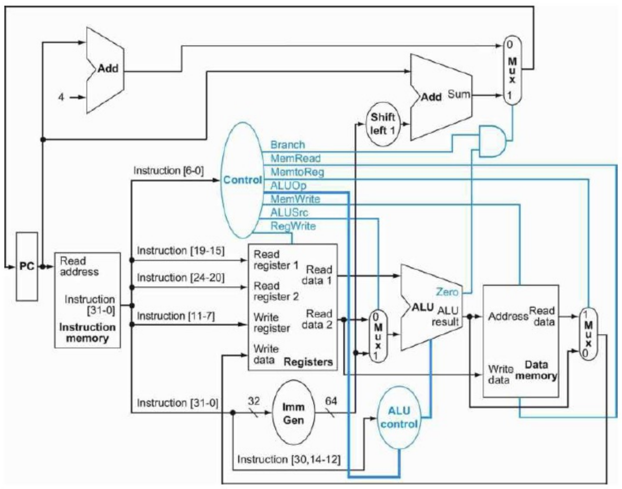
Solution:

```
f:
     addi sp,sp,-12      # adjust stack for 3 items
     sw x20, 8(sp)       # save x20 for use afterwards
     sw x21, 4(sp)       # save x21 for use afterwards
     sw x22, 0(sp)       # save x22 for use afterwards

     lw   x20, 4(x6)     # x20 = A[1]
     add  x11, x11, x7   # x11 = &B[i]
     lw   x21,0(x11)     # x21 = B[i]
     add  x21,x21,x20    # x21 = x21+x20 = B[i]+A[1]
     slli x21, x21, 2    # x21 = x21*4
     add  x21,x21, x7    # x21 = &A[B[i]-A[1]]
     lw   x22, 0(x21)    # x22 = A[B[i]+A[1]]
     add  x10,x10, x22   # return value g + A[B[i]-A[1]];

     lw x22, 0(sp)       # restore register x22 for caller
     lw x21, 4(sp)       # restore register x21 for caller
     lw x22, 8(sp)       # restore register x22 for caller
     addi sp,sp,12       # adjust stack to delete 3 items
     jalr x0,0(x1)       # jump back to calling routine
```

3. Question - Instruction execution (10 marks) In this exercise, we examine in detail how an instruction is executed in a single-cycle datapath.



Problems in this exercise refer to a clock cycle in which the processor fetches the following instruction:

For the instruction

`sw x12, 11(x13)`

a) (2 marks) What are the values of the ALU control unit's inputs for this instruction?

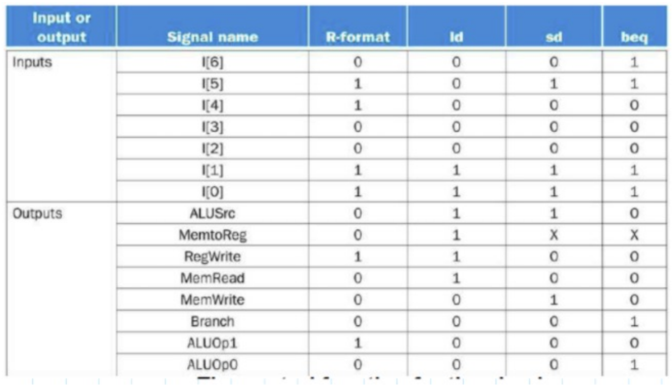
b) (2 marks) What is the new PC address after this instruction is executed?

c) (2 marks) For each mux, show the values of its inputs and outputs during the execution of this instruction. List values that are register outputs at `Reg [`x_n`]`.

d) (2 marks) What are the input values for the ALU and the two add units?

e) (2 marks) What are the values of all inputs for the registers unit?

a) ALU control unit's inputs:
ALU control lines: 0010
(30,12-14) 0 010
ALU op = 00 from the table for "sw"



Input or output	Signal name	R-format	ld	sd	beq
Inputs	I[6]	0	0	0	1
	I[5]	1	0	1	1
	I[4]	1	0	0	0
	I[3]	0	0	0	0
	I[2]	0	0	0	0
	I[1]	1	1	1	1
	I[0]	1	1	1	1
Outputs	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

b) The new PC is the old PC + 4. Th is signal goes from the PC, through the "PC + 4" adder, through the "branch" mux, and back to the PC.

c) ALUsrc : Inputs: Reg[x12] and 0x0000000B ; Output: 0x0000000B
MemToReg : Inputs: Reg[x13] + 0x0B and <undefined> ; output: <undefined>
Branch: Inputs: PC + 4 and PC + 0x00000016

d) ALU inputs: Reg[x13] and 0x0000000B
PC + 4 adder inputs: PC and 4

Branch adder inputs: PC and 0x00000016

e) Read register 1 - 01101 , instruction [19-15]
Read register 2 - 01100 , instruction [24-20]
Write Register: - 01011, instruction [11-7]
Write Data ⟨undefined⟩

4. Question (10 marks) The processor fetches the following instruction word:
0000 0000 1011 0110 0010 0100 0010 0011
a) (2 marks) What are the values of "opcode", "funct", "imm" and "registers" ?

b) (2 marks) What is instruction type?

c) (2 marks) Describe the instruction using formal notation (in Verilog). See Risc - V Reference data for description in Verilog, or Chapter 2 slide 78.

d) (2 marks) Describe instruction in plain words

e) (2 marks) Encode instruction `lw x9, 10(x24)`

Solution
a)
opcode: 010 0011 -
imm[4:0]: 0100 0
func3: 010 -
rs1: 0110 0 = 12
rs2: 0 1011 = 11
imm[11:5]: 0000 000

Encoded instruction is:
`sw x11, 8(x12)`

b) S type

c) M[R[rs1] + imm](31:0) = R[rs2](31:0)

d) Copy content of rs2 to memory location rs1+imm
e)
0x00ac2483
0000 0000 1010 1100 0010 0100 1000 0011

5. Question - Memory Hierarchy (10 marks)

Assume we have byte addressable memory and 32-bit address.
a) (3 marks)
For direct-mapped cache of 8KB in size, two words (8 bytes) per block calculate the size of:
Tag, Block index, Word and Byte offset

b) (2 marks) Into what block bytes from the addresses 0x00003333 would be stored ?

c) (3 marks) For a 4-way set associative cache with cache size of 2048 blocks, 8 word block size calculate the size of:
Tag, Index, Word and Byte offset
d) (2 marks) Into which set bytes from the addresses 0x00003333 would be be stored?

Solution
a)
8KB cache size is 8192 bytes. We have two words (8 bytes) per block, so we have 1024 blocks and we need 10 bits for block index. For byte offset we need 3 bits, and for Tag (32-10-3 = 19)
b)
13107 modulo 1024 = 819
c)
4-way set associative uses 4 blocks per set, thus cache uses 512 sets or

9-bit index field.
8 words per block so thats 3-bit word offset field and 2-bit byte offset.
Tag size = 32 - 9 - 3 - 2 = 18
d)
13107 modulo 512 = 307 set

6. Bonus Question (2 marks) Consider a SEC code that protects 8 bit words with 4 parity bits. If we read the value 0x385
a) (1 mark) Represent the value (0x385) as 12 bit encoded word and indicate data and parity bits.

b) (1 mark) Is there an error? If so, correct the error.

Solution
a) We can represent 0x385 as

p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8
0	0	1	1	1	0	0	0	0	1	0	1

0x385
b) Is there an error? If so, correct the error.
$p1 = 0 = d1 \text{ or } d2 \text{ or } d4 \text{ or } d5 \text{ or } d7 = 1 \text{ or } 1 \text{ or } 0 \text{ or } 0 \text{ or } 0 = 0$
$p2 = 0 = d1 \text{ or } d3 \text{ or } d4 \text{ or } d6 \text{ or } d7 = 1 \text{ or } 0 \text{ or } 0 \text{ or } 1 \text{ or } 0 = 0$
$p4 = 1 = d2 \text{ or } d3 \text{ or } d4 \text{ or } d8 = 1 \text{ or } 0 \text{ or } 0 \text{ or } 1 = 0 <- \text{error}$
$p8 = 0 = d5 \text{ or } d6 \text{ or } d7 \text{ or } d8 = 0 \text{ or } 1 \text{ or } 0 \text{ or } 1 = 0$

p4 is 1, should be 0

p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8
0	0	1	0	1	0	0	0	0	1	0	1

0x285