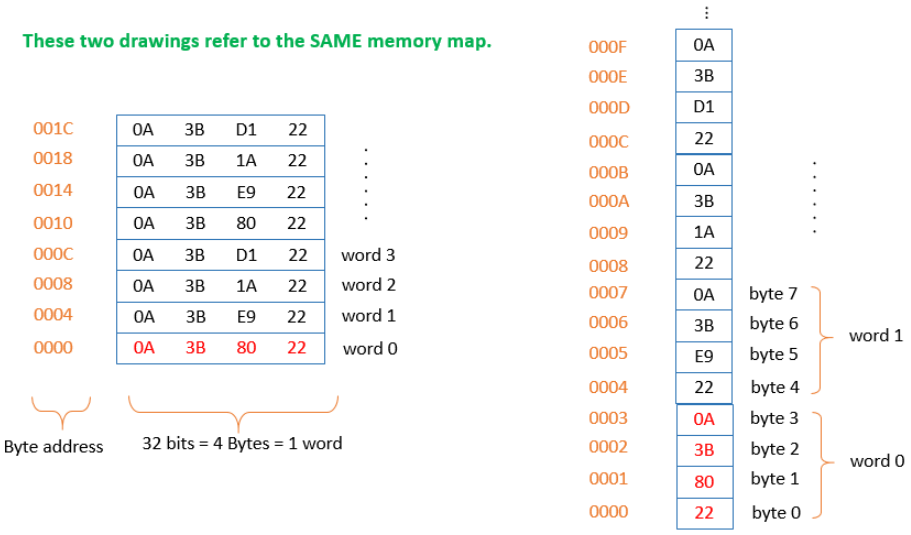
COMP 1047 Lab Week 06

MIPS Instructions and Single Cycle CPU

(1) This part aims to provide with you deeper insights into the memory organization and representation. Observe the figure below for the same piece of memory, which is represented in words and in bytes, respectively.



Based on the figure, provide answers for the questions below.

(a) What is the byte address of word number 42? Can you represent the address in Hex format?

Solution Word 0 starts at 0 (the initial byte), Word 1 starts at 4, etc. Keeping counting, Word 42 starts at $42 \times 4 = 168$, which is A8 in Hex.

(b) What are the byte addresses that word 42 spans?

Solution 0xA8, 0xA9, 0xAA, 0xAB.

(c) Write the MIPS assembly code to load word 3 into `$t0`. Hint: (1) Use `lw`; (2) Use `$t1` to contain the address of word 3 directly.

Solution

```
addi $t1, $0, 12 (or $t1, $0, 0xC)
lw $t0, 0($t1)
```

(d) Write the MIPS assembly code to add the values in word 0 and word 1, and store the result back to word 42.

Solution

```
lw $t6, 0($0)
addi $t1, $0, 4
lw $t7, 0($t1)
add $t8, $t6, $t7
addi $t1, $0, 0xA8
sw $t8, 0($t1)
```

(2) Translate the instruction `add $s0, $s1, $s2` into machine code, and represent it into Hex format. Hint – The following info is provided: opcode = 0, funct = 32. For register numbering, please check it in the lecture notes, or Table 6.1 in the Harris & Harris book (ed1). Note: During examination, either the above info will be directly given, or Tables B.1 and B.2 in the Harris & Harris book (ed1) will be provided to answer this type of question.

Solution 0x0232 8020

(3) Translate the instruction `addi $t0, $t6, -5` into machine code, and represent it into Hex format. Hint – The following info is provided: opcode = 8.

Solution 0x21C8 FFFB

(4) Translate the instruction `lw $t2, 32($0)` into machine code, and represent it into Hex format. Hint – The following info is provided: opcode = 35.

Solution 0x8C0A 0020

(5) Practice the following translations of the instructions `sw, sub, lb, sb, and, or, sll, srl, sra.`

(6) Design an ALU which takes in two 32-bit input A and B, and drives one 32-bit output Y. The ALU should support the following operations: add, sub, and, or, slt, and "identical bit checking". Identical bit checking means to bit-wisely check A and B: At each bit position, if the corresponding bits of A and B are the same, then output 1 at that position. Otherwise 0 is output. Example: with A = 1001 and B = 0101, the output of this operation is 0011. Design the ALU using the logic blocks mentioned in the lecture notes, and provide the corresponding ALUop control signal for each operation (i.e., provide a control table similar to the one in Slide 20 in the 'ALU Design' lecture notes.).

(a) Provide the design if ALUop is required to be 4 bits.

(b) Provide the design if ALUop is required to be 3 bits.

Solution See Figure 1 and Figure 2.

(7) This question is about datapath design of the single cycle CPU. Recall that in Page 27 of this week's lecture notes ("Single-Cycle CPU"), an animation of the data flow process of the `add` instruction has been illustrated, as shown below. Your task is to provide similar animations for the `lw, sw` and `beq` instructions.

Resources provided for assistance:

- A ppt template (pp. 27) extracted for you to design your own animation on. Available in Section "Week 6" in the Moodle page.
- We had a MIPS online learning & simulator tool designed by a group of your senior classmates taken SYS in AY2019-2020. [Wonderful job indeed!] Play with it, and use it as the guidance and reference for your design. The link is here. The user manual is here. Something for fun is here.

Figure 1 Sample solution 4(a)

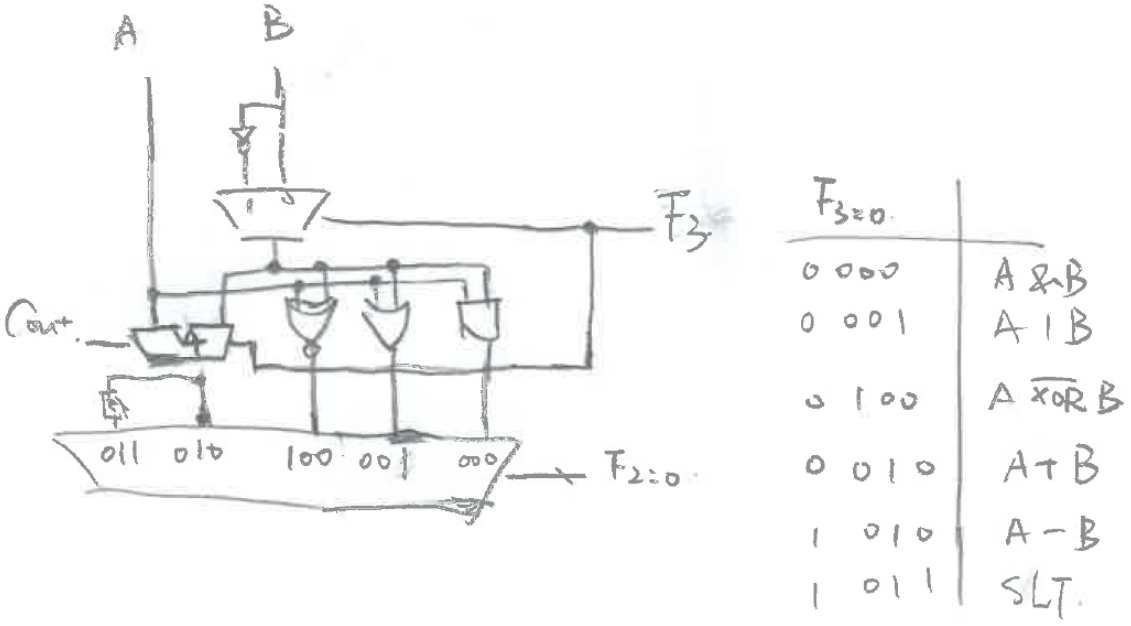
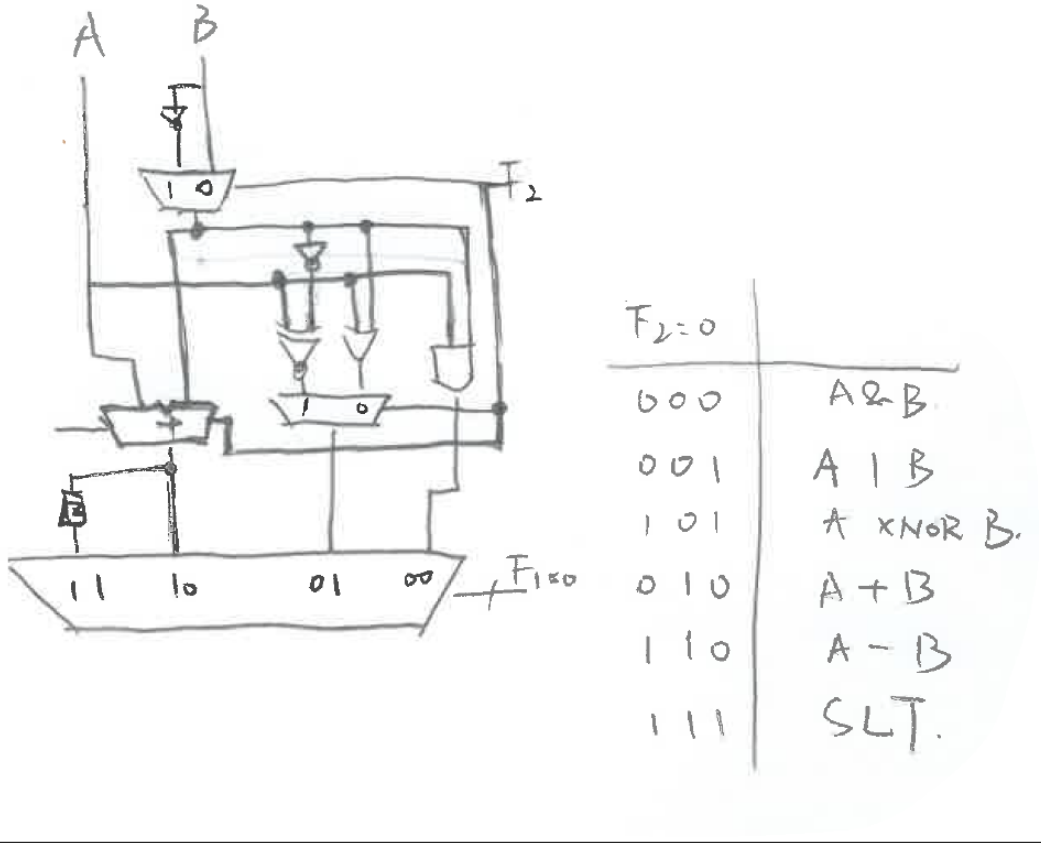


Figure 2 Sample solution 4(b)



University of Nottingham
UK | CHINA | MALAYSIA

Building the Datapath

Dataflow during **add**