# Homework 13

ECE2504 CRN:82729

Jacob Abel

December 5, 2017

**Question 1:** (8 pts) Specify the 20-bit control word that must be applied to the datapath of the single-cycle simple computer in Chapter 8 of your textbook (See Fig 8-16) to implement each of the following microoperations. Use the control word fields shown in Figure 8-16. Present your answers in table form.

| Operation | $DA$ | $AA$ | $BA$ | $MB$ | $FS$ | $MD$ | $RW$ | $MW$ | $PL$ | $JB$ | $BC$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| a) $R2 \leftarrow$ sr Constant in | 010 | xxx | cnst | 1 | 1101 | 0 | 1 | 0 | 0 | 0 | 1 |
| b) $M[R4] \leftarrow R2$ | xxx | 100 | 010 | 0 | 0000 | 1 | 0 | 1 | 0 | 0 | 0 |
| c) $R1 \leftarrow R3 \vee$ Constant in | 001 | 011 | cnst | 1 | 1001 | 0 | 1 | 0 | 0 | 0 | 1 |
| d) $R4 \leftarrow R3 - 1$ | 100 | 011 | 001 | 0 | 0110 | 0 | 1 | 0 | 0 | 0 | 0 |

**Question 2:** (8 pts) Given the following 16-bit control words for the Simple Computer datapath shown in Fig. 8-11, determine

- the microoperation sequence that is executed (in RTL),
- the change in register contents

Consider each control word separately, i.e. this is not a sequence. Assume the initial register contents shown is the starting point for each of these microoperations. Assume that Constant = 6 and Data_in = 0x1B

| Initial Register Contents | |
|---|---|
| $R0$ | 00000000 |
| $R1$ | 00100000 |
| $R2$ | 01000100 |
| $R3$ | 01000111 |
| $R4$ | 01010100 |
| $R5$ | 01001100 |
| $R6$ | 01000001 |
| $R7$ | 01001001 |

a) 101 100 101 0 1000 0 1

$R5 \leftarrow R5$

| Updated Register Contents | |
|---|---|
| $R0$ | 00000000 |
| $R1$ | 00100000 |
| $R2$ | 01000100 |
| $R3$ | 01000111 |
| $R4$ | 01010100 |
| $R5$ | 01000100 |
| $R6$ | 01000001 |
| $R7$ | 01001001 |

b) 110 010 100 0 0101 0 1

$R6 \leftarrow R2 - R4$

| Updated Register Contents | |
|---|---|
| $R0$ | 00000000 |
| $R1$ | 00100000 |
| $R2$ | 01000100 |
| $R3$ | 01000111 |
| $R4$ | 01010100 |
| $R5$ | 01001100 |
| $R6$ | 11110000 |
| $R7$ | 01001001 |

c) 101 110 000 0 1100 0 1

$R5 \leftarrow R4 \wedge R5$

| Updated Register Contents | |
|---|---|
| $R0$ | 00000000 |
| $R1$ | 00100000 |
| $R2$ | 01000100 |
| $R3$ | 01000111 |
| $R4$ | 01010100 |
| $R5$ | 01001100 |
| $R6$ | 01000001 |
| $R7$ | 01001001 |

d) 101 000 000 0 0000 0 1

$R5 \leftarrow R0$

| Updated Register Contents | |
|---|---|
| $R0$ | 00000000 |
| $R1$ | 00100000 |
| $R2$ | 01000100 |
| $R3$ | 01000111 |
| $R4$ | 01010100 |
| $R5$ | 00000000 |
| $R6$ | 01000001 |
| $R7$ | 01001001 |

**Question 3:** (4 pts) A 32-bit computer has a memory unit and a register file with 64 registers. The instruction set consists of 68 different operations. There is only one instruction format with an opcode, a register operand, and an immediate operand. Each instruction is stored in a single word of memory.

a) How many bits are required for the opcode? 7
b) How many bits are left for the immediate operand? 19
c) If the immediate operand is used as an unsigned address to memory, what is the maximum number of words that can be addressed in memory? 524288
d) What is the range of signed immediate operands that can be accommodated? -262144 : 262143

**Question 4:** (4 pts) Give an instruction for the single cycle computer that resets register $R4$ to 0 and updates the $Z$ and $N$ status bits based on the value 0 loaded into $R4$. By examining the ALU logic provided in Chapter 8, determine the values of the $V$ and $C$ status bits.

| Instruction | Operation | Opcode | V | C |
|---|---|---|---|---|
| ANDI R5 R5 0 | $R5 \leftarrow R5 \wedge 0$ | 1001000101101000 | 0 | 0 |

**Question 5:** (18 pts) Consider the following sequence of assembly language instructions for the Single Cycle Computer show in section 8-8 of your textbook. Assume that the initial register contents are equal to each register's index (i.e. $R0$ contains 0, $R1$ contains 1, $R2$ contains 2, etc). Provide your results in table form.

a) Give the machine instructions in hex in the same row of the table as the assembly instruction.
b) Give the contents of any register changed by the instruction, or the location and contents of any memory location changed by the instruction, in the next row of the table.

Note: the information is positioned in this way because new values (due to the execution of an instruction) do not appear in registers or memory until after a positive clock edge has occurred.

| | |
|---|---|
| add r0, r1, r2 | 0x040A |
| $R0 = 3$ | |
| sub r3, r4, r5 | 0x0AE5 |
| $R3 = -1$ | |
| sub r6, r7, r0 | 0x0BB8 |
| $R6 = 10$ | |
| add r0, r0, r3 | 0x0403 |
| $R0 = 2$ | |
| sub r0, r0, r6 | 0x0A06 |
| $R0 = -4$ | |
| st r7, r0 | 0x4038 |
| $M7 = -4$ | |
| ld r7, r6 | 0x21F0 |
| $R7 = M10$ | |
| adi r0, r6, 2 | 0x8432 |
| $R0 = 12$ | |
| adi r3, r6, 3 | 0x84F3 |
| $R3 = 15$ | |

**Question 6:** (8 pts)
Design a 8x6 ROM with the following content

| Address | Data |
|---|---|
| 000 | 111000 |
| 001 | 000001 |
| 010 | 000000 |
| 011 | 100000 |
| 100 | 001110 |
| 101 | 111111 |
| 110 | 010011 |
| 111 | 110110 |

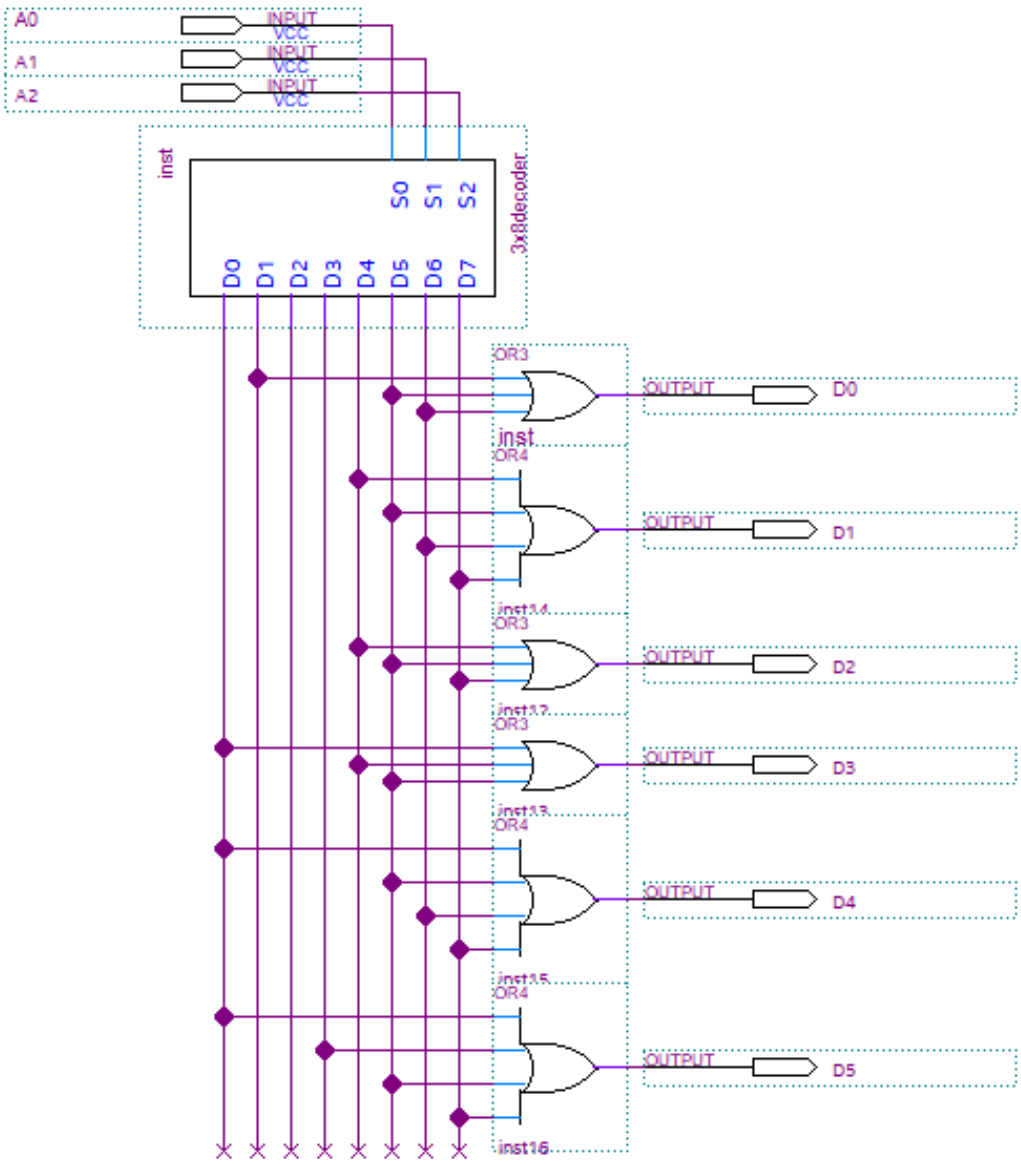


Figure 1: 8x6 ROM

GRADING SCALE

Total: 50 pts

| Pts | 0 | 6 | 12 | 18 | 25 | 31 | 37 | 43 |
|---|---|---|---|---|---|---|---|---|
| Letter Grade | D- | D | C- | C | B- | B | A- | A |