**Fall 2008, CS 4341.002 (Honors), Digital Logic and Computer Design**

**Project**

**G. R. Dattatreya**

**Due BEFORE class on Monday, November 17, 2008**

In this project you will design a specific hardware system from fundamental principles. The approach should be to design the first level building blocks from gates. The second and succeeding level building blocks should be developed with the help of lower level building blocks. Think of the gates as the zeroth level building blocks. In each case, the inputs, outputs, timing, and functions of each of the used and developed (or being developed) building blocks must be clear. With this approach, the bigger functional blocks will not look cluttered.

The Datapath will have an ALSU (Arithmetic, Logic, Shift Unit), registers, a memory (RAM) unit, Input/Output ports, data transfer buses/connections, microoperation enable lines, data input and output connections between the system and the control unit. You will not be designing the control unit. Data words will be 16 bits long, throughout. We will have one word input terminal from the control unit to the Datapath and one word output to the control unit, from the Datapath.

We will have 13 identical registers $R_0, ..., R_{12}$, each 16 bits long, in the Data Path. An additional data input to the system is like a fictitious register $R_{13}$. Similarly, a data destination to the outside is like another fictitious register $R_{14}$.

The ALSU will have 16 microoperations, exactly one of which will be active during any time. Let $B$ and/or $A$ be the generic name/s of the input operand/s to the ALSU. Notice that each of $A$ and $B$ can be one of the 13 registers or the Input from the control unit. The arithmetic operations are in 2's complement notation. The entire ALSU is a combinational system. The $C_{in}$ comes in from outside. There are four status flags which are functions of the result of the current ALSU operations. They are the $V$ for overflow, $Z$ which should be 1 if the result turns out to be all zero bits, $C$, which is the Carry out of the sign position, and $S$, the sign bit of the result. These four lines go into the control unit. Following are the ALSU operations.

| Serial number | Acronym | Function | Comment | Flags |
|---|---|---|---|---|
| 1 | TRA | $A$ | Output $A$ | $ZS$ |
| 2 | ADD | $A+B$ | | $VCZS$ |
| 3 | NEG | $-B$ | Output negative of $B$ | $VCZS$ |
| 4 | SUB | $A-B$ | | $VCZS$ |
| 5 | INC | $A+1$ | unsigned | $ZS$ |
| 6 | DEC | $A-1$ | unsigned | $ZS$ |
| 7 | SHR | Logic Shift Right $0, A$ | with 0 shifted in | $CZS$ |
| 8 | DIV | $\frac{A}{2}$ | | $CZS$ |
| 9 | MPY | $A \times 2$ | | VCZS |
| 10 | SHL | Logic Shift Left $A, 0$ | with zero shifted in | $CZS$ |
| 11 | SLC | Shift left $A, C_{in}$ | with $C_{in}$ | $CZS$ |
| 12 | SHR | Shift right $C_{in}, A$ | with $C_{in}$ | $CZS$ |
| 13 | AND | $A \cdot B$ | bit by bit AND | $ZS$ |
| 14 | IOR | $A+B$ | bit by bit OR | $ZS$ |
| 15 | INV | $\overline{A}$ | | $ZS$ |
| 16 | NOP | Hi-impedance | Release output bus | |

The memory (RAM) has 64K words, each 16 bits long. Any register can act as the address register and any register can be used to provide input or get output from the RAM. The first 32 of these 64K words are actually the memory mapped Input/Output ports. This means that there is an additional bus connecting these 32 words to the outside world. Outside world can read out or write into any of these words, irrespective of what the rest of the system is doing. The additional address and data buses are also part of the Datapath.

Additional clarifications will be given as topics are covered in class. You should start with the design of the ALSU. You can then separately design the registers, the bus transfer systems, and then the RAM including the I/O ports.