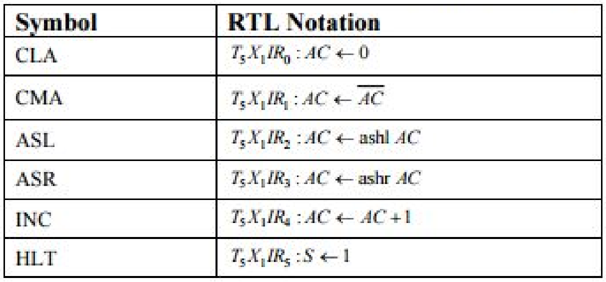
# CEG2136 Lab 4

*Brennan McDonald* - *8195614*
12/07/2016

## *Introduction*

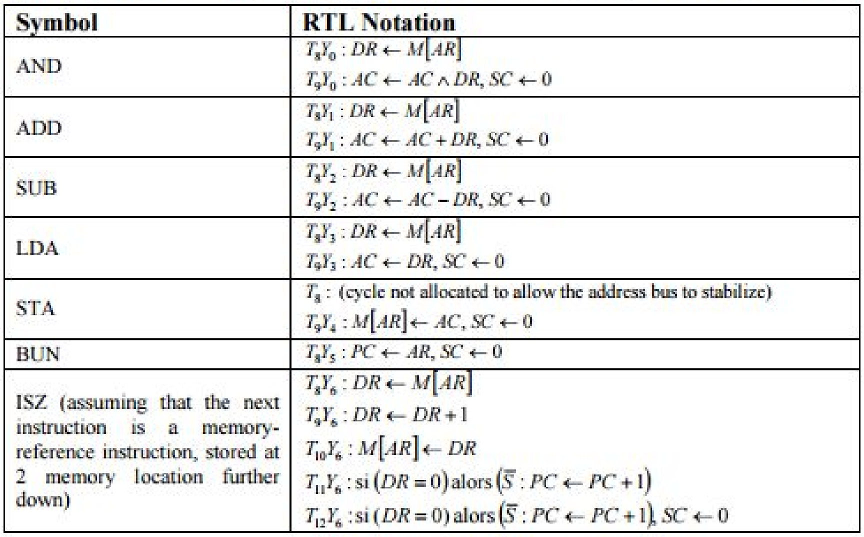
The purpose of this lab was to observe the the processing unit of a basic computer and to derive/implement equations for the control unit of a basic computer. In this lab students were also told to create a basic program in an assembly instruction set that the was predefined. To solve this problem we were required the basic layouts of what our processor was supposed to do. This information was acquired from the following charts:

**Table 3: Instruction Execution Cycle - Control of the register - reference instructions**



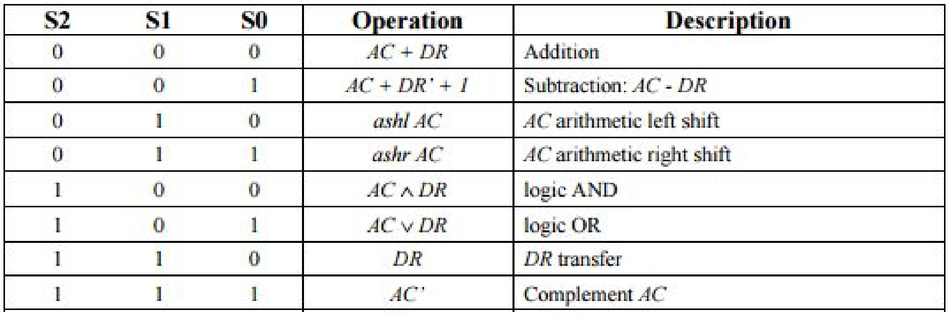
| Symbol | RTL Notation |
|---|---|
| CLA | $T_5X_1IR_0 : AC \leftarrow 0$ |
| CMA | $T_5X_1IR_1 : AC \leftarrow \overline{AC}$ |
| ASL | $T_5X_1IR_2 : AC \leftarrow \text{ashl } AC$ |
| ASR | $T_5X_1IR_3 : AC \leftarrow \text{ashr } AC$ |
| INC | $T_5X_1IR_4 : AC \leftarrow AC + 1$ |
| HLT | $T_5X_1IR_5 : S \leftarrow 1$ |

**Table 4: Instruction Execution Cycle -** Control of the **memory - reference instructions**



| Symbol | RTL Notation |
|---|---|
| AND | $T_8Y_0 : DR \leftarrow M[AR]$<br>$T_9Y_0 : AC \leftarrow AC \wedge DR, SC \leftarrow 0$ |
| ADD | $T_8Y_1 : DR \leftarrow M[AR]$<br>$T_9Y_1 : AC \leftarrow AC + DR, SC \leftarrow 0$ |
| SUB | $T_8Y_2 : DR \leftarrow M[AR]$<br>$T_9Y_2 : AC \leftarrow AC - DR, SC \leftarrow 0$ |
| LDA | $T_8Y_3 : DR \leftarrow M[AR]$<br>$T_9Y_3 : AC \leftarrow DR, SC \leftarrow 0$ |
| STA | $T_8$ : (cycle not allocated to allow the address bus to stabilize)<br>$T_9Y_4 : M[AR] \leftarrow AC, SC \leftarrow 0$ |
| BUN | $T_8Y_5 : PC \leftarrow AR, SC \leftarrow 0$ |
| ISZ (assuming that the next instruction is a memory-reference instruction, stored at 2 memory location further down) | $T_8Y_6 : DR \leftarrow M[AR]$<br>$T_9Y_6 : DR \leftarrow DR + 1$<br>$T_{10}Y_6 : M[AR] \leftarrow DR$<br>$T_{11}Y_6 : \text{si } (DR = 0) \text{ alors } (\overline{S} : PC \leftarrow PC + 1)$<br>$T_{12}Y_6 : \text{si } (DR = 0) \text{ alors } (\overline{S} : PC \leftarrow PC + 1), SC \leftarrow 0$ |

**Table 5: ALU operations table**



| S2 | S1 | S0 | Operation | Description |
|---|---|---|---|---|
| 0 | 0 | 0 | $AC + DR$ | Addition |
| 0 | 0 | 1 | $AC + DR' + 1$ | Subtraction: $AC - DR$ |
| 0 | 1 | 0 | $ashl\ AC$ | $AC$ arithmetic left shift |
| 0 | 1 | 1 | $ashr\ AC$ | $AC$ arithmetic right shift |
| 1 | 0 | 0 | $AC \wedge DR$ | logic AND |
| 1 | 0 | 1 | $AC \vee DR$ | logic OR |
| 1 | 1 | 0 | $DR$ | $DR$ transfer |
| 1 | 1 | 1 | $AC'$ | Complement $AC$ |

The solution to the presented problem was to look through the previous charts and extract equations for each of the micro-operations of the processor. These equations were then implemented in the Quartus software and uploaded to the Altera Board.

*Design*

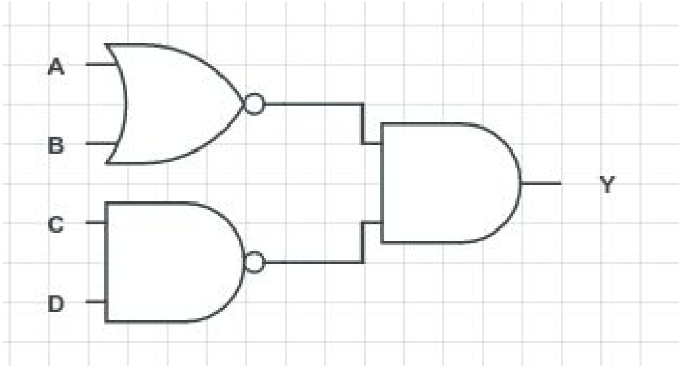


*Fig 1.*

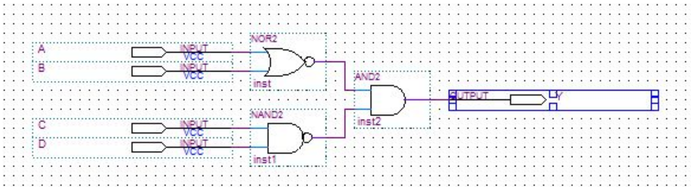
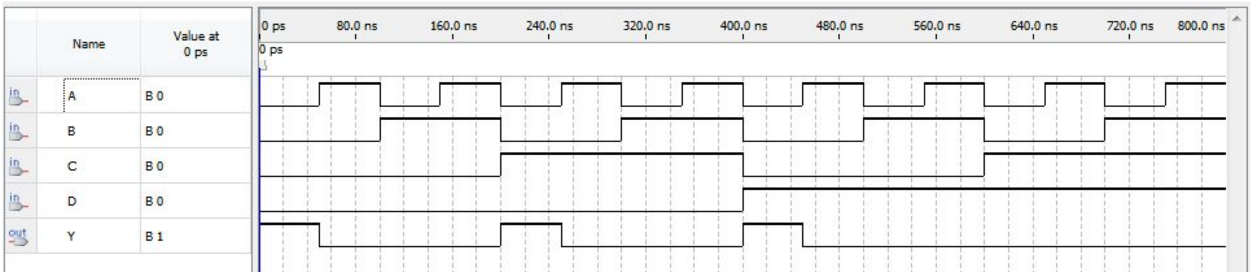


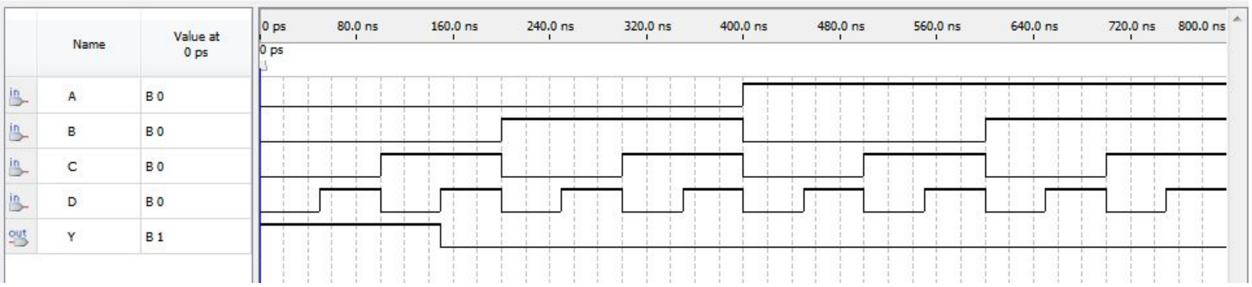
*Fig 2*

*Simulation and Verification*



| Name | Value at 0 ps |
|---|---|
| A | B 0 |
| B | B 0 |
| C | B 0 |
| D | B 0 |
| Y | B 1 |

*Fig. 4. Waveform Diagram with A as the least significant bit*



| Name | Value at 0 ps |
|---|---|
| A | B 0 |
| B | B 0 |
| C | B 0 |
| D | B 0 |
| Y | B 1 |

*Fig. 4. Waveform Diagram with A as the most significant bit*

## *Discussion*

Due the the time constraints and busy nature of the end of the semester I was not able to submit the code part of my lab and therefore was not able to verify that it worked correctly. However due to the ease of the equations the hardware section of the lab was simple to follow and implement.

*<u>Prelab</u>*

2 $(\overline{A+B})(\overline{CD})$

3

NAND:

| A | B | out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

NOR:

| A | B | out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

4

| A | B | C | D | out |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |