Matthew Cloutier

ECE 505

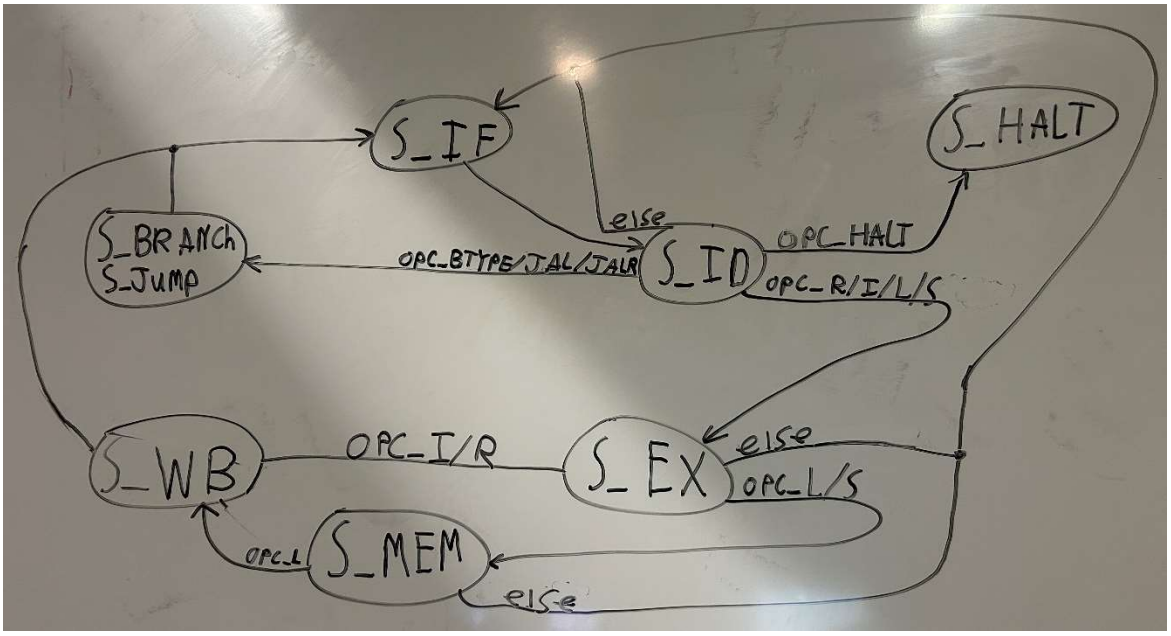
Multi Cycle Processor Project

Description of Project

During this project a multi cycle processor was developed based on the previous design of a single cycle processor. The main difference was the control unit, which utilizes a state machine to control the current function of the processor. To conclude the project the multi cycle and single cycle processors were compared.

FSM Controller Design

The FSM controller controls the control signals depending on the current state of the FSM. These states are IF, ID, EX, MEM, WB, JUMP/BRANCH, and HALT. Below is a state machine that describes the transitions between these states:



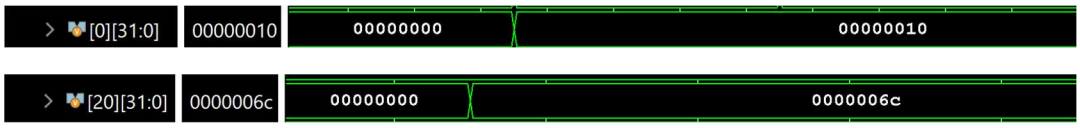
Top-Level Processor Integration and Verification

After designing the FSM controller, it was implemented into the top module replacing the single-cycle controller. To allow it to function two modifications were

necessary. The first of which was updating wd, the register being written to, at the start of each clock cycle as opposed to it being combinational. The second change was creating a mem_data_reg, which stores the output of the memory for one extra clock cycle. These two changes were both necessary because the data had to survive for one more cycle each because of the nature of a multi cycle processor.

This design was verified using both program 1 and program 2, which both ran successfully. Below are screenshots showing the final values of the relevant memory, which prove the correct function of the processor.

Program 1 Relevant Memory



These two screenshots show the final two instructions of program 1, sw x5 0(x1) and sw x6 4(x2). Both values and addresses in memory match the simulation of the code, proving the correct function of the processor.

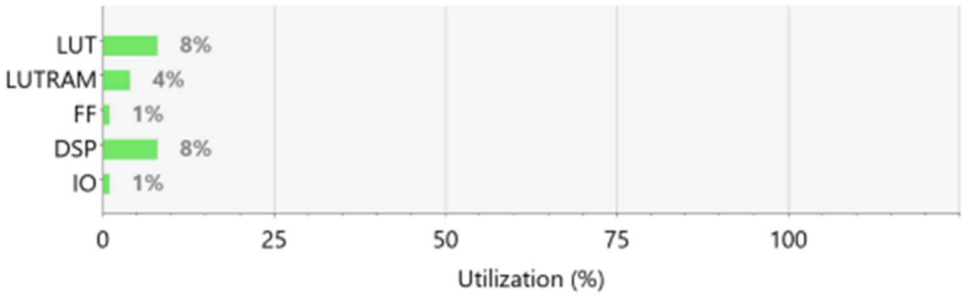
Program 2 Relevant Memory

This screenshot shows the transition of memory address 12 for the last instruction of program 2, which stores x18 at 0(x5). This result matches with simulation of the code, proving correctness of the processor.

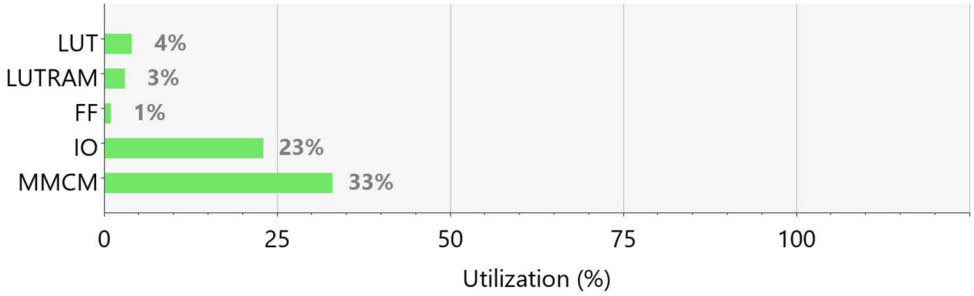
Comparing to Single Cycle

Utilization

Comparing the utilization between the multi cycle processor created here and the previously created single cycle processor, the multi cycle processor utilized less resources. The only resource the multi cycle used more of was the MMCM. For the utilization, of the multi cycle the IO should be ignored when comparing them. The IO usage is much higher on the multi cycle because mem_data was set an output to force the implementation to implement everything, and a different strategy was used on the single cycle. The two screenshots are shown below:



Single Cycle Utilization



Multi Cycle Utilization

Timing

The worst negative slack for the multi cycle is 0.557 ns, which translates to being able to run the processor at a maximum speed of roughly 100 MHz. Comparing this frequency to the 20 MHz benchmark of the single cycle processor it is 5 times faster than the single cycle. All five cycles are not used every time either, making this multi cycle processor faster than the single cycle CPU.