Matthew Liu
Lab Report 6
ECE 2031 L09
3 March 2019

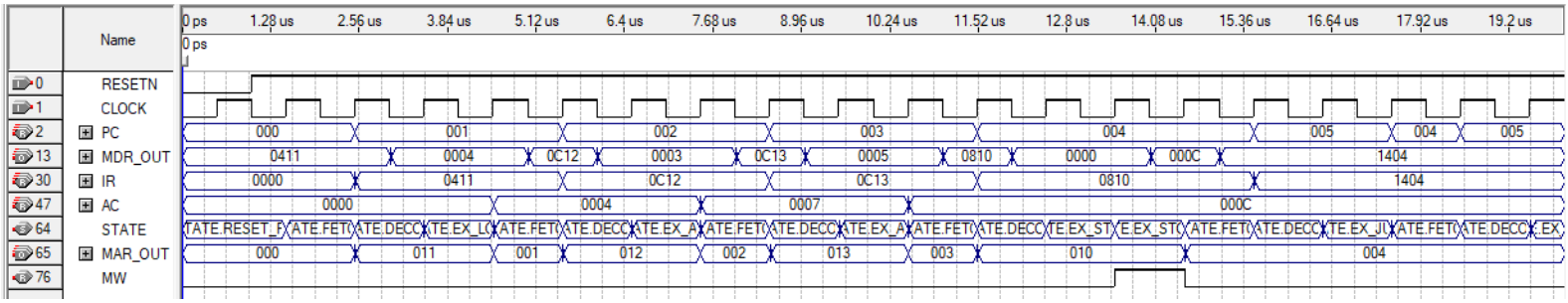

Figure 1. Simple Computer simulation waveform for edited example.mif (A = (B + C) + D). The waveform's machine code comes from the edited code program example.asm which is assembled with SCASM to create a mif file. The result is then functionally simulated in Quartus with an input waveform file created in the lab.

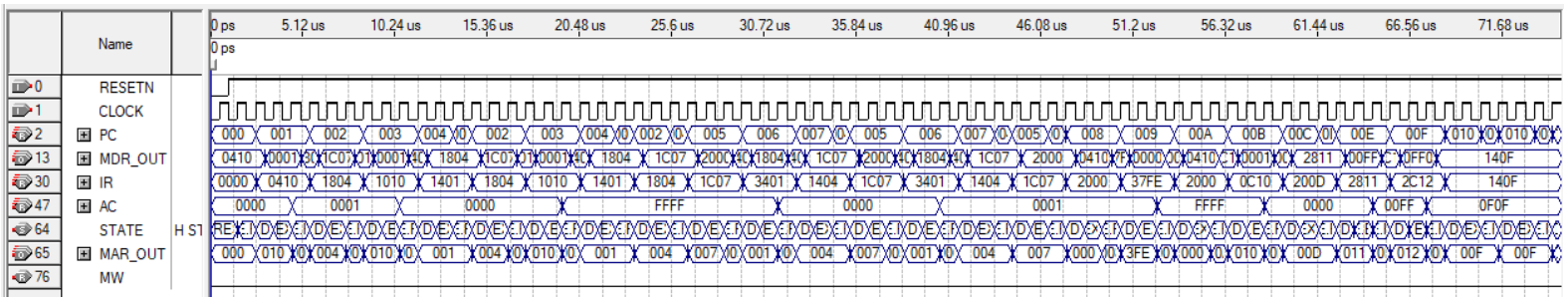

Figure 2. Test_code.mif program file's timing simulation waveform - downloaded from class website. The waveform is created after adding the SUB, JNEG, JPOS, JZERO, OR, XOR, and ADDI instructions to the simple computer as defined by Table 7.1 in the lab. (Note: After compilation of program, reaching state 0F0F is indicative of reaching successful state transitions throughout the program output).

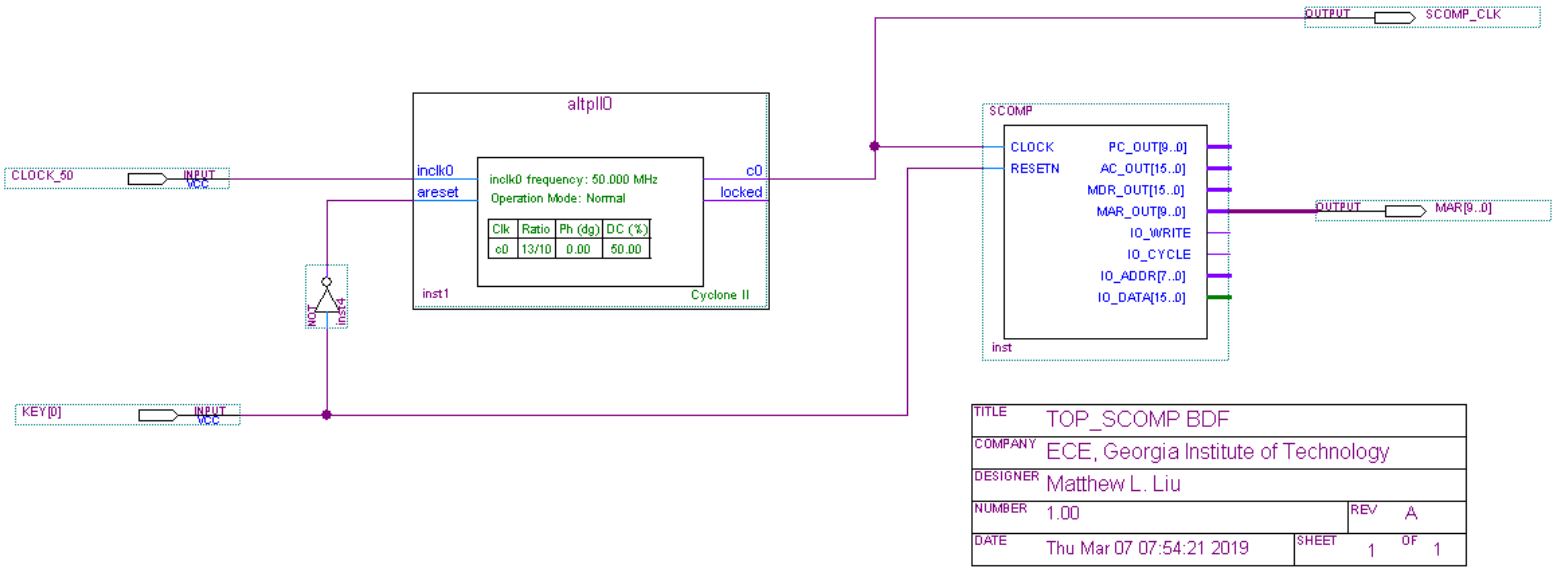

Figure 3. Schematic (TOP_SCOMP.BDF) – part of SCOMP.QPF project, used for the DE2 Cyclone II chip. The schematic includes the SCOMP.BSF symbol created from SCOMP.VHD file which includes all the instructions to the Simple Computer defined by Table 7.1 in the lab. Altpll() device is used for clocking input for SCOMP (notes: measured Max Clock frequency 67.05 MHz, Key[0] push button resets both devices, and clock_50 is a 50 MHz DE2 clock).

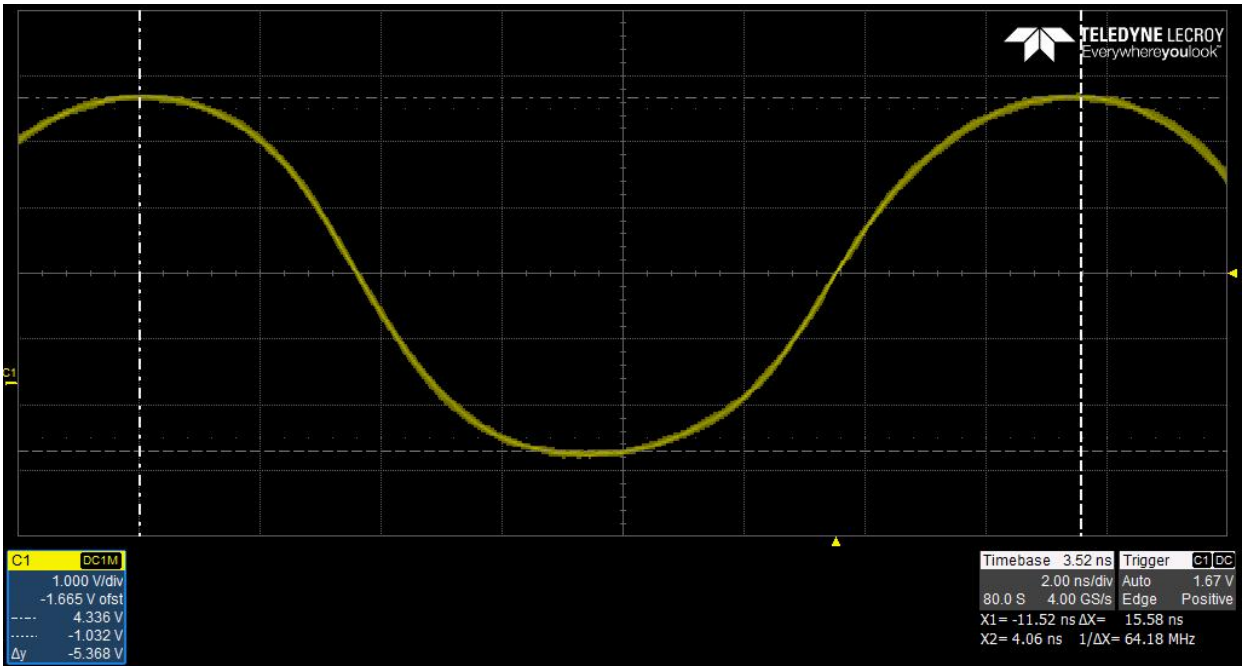

Figure 4. Oscilloscope waveform measurement from DE2 board. The waveform simulates the Quartus project including the atpll() and SCOMP device created in the lab. The purpose of this is to measure he frequency of the clock with the oscilloscope to verify what we expect. Note that the measured frequency is 64.18 MHz, which is just slightly lower than the set Max Clock Frequency (65 MHz) set by atpll().

APPENDIX A
LAB 7 DESIGNING SIMPLE COMPUTER

```
-- example.mif (edited) A = (B+C) + D
-- Lab 7 SCOMP Lab, example.asm/mif file
-- Altera Memory Initialization File (MIF) for Pre-lab
-- Matthew Liu
-- ECE2031 L09
-- 7 March 2019

DEPTH = 1024;
WIDTH = 16;

ADDRESS_RADIX = HEX;
DATA_RADIX = HEX;

CONTENT
  BEGIN
    [000..3FF] : 0000;  -- Default to NOP

        000 : 0411;  -- Start:   LOAD      B          ;Load value stored in B
        001 : 0C12;  --          ADD       C          ;Add value stored in C
        002 : 0C13;  --          ADD       D          ;Add value stored in D
        003 : 0810;  --          STORE     A          ;Store/replace value back in A
        004 : 1404;  -- Here:    JUMP      Here       ;Loop here forever
        010 : 0000;  -- A:       DW        &H0000
        011 : 0004;  -- B:       DW        &H0004
        012 : 0003;  -- C:       DW        &H0003
        013 : 0005;  -- D:       DW        &H0005     ;New Variable D
  END;
```