ECSE 318 Lab 2 Report

Group 5

Nabeeh Daouk and Kyle Heston

Sunday, October 8, 2023

Lab Part 1_ ALU:

```
# Testing operation:       ADD=00000    //A+B=>C    signed addition
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0014                vout:1   cout:1  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000010100    vout:1   cout:1  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    20              vout:1   cout:1  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    20              vout:1   cout:1  |
# _______________________________________________________________________________________________________________________________________
# Testing operation:      ADDU=00001   //A+B=>C    unsigned addition
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0014                vout:0   cout:1  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000010100    vout:0   cout:1  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    20              vout:0   cout:1  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    20              vout:0   cout:1  |
# _______________________________________________________________________________________________________________________________________
# Testing operation:       SUB=00010    //A-B=>C    signed subtraction
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0010                vout:0   cout:1  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000010000    vout:0   cout:1  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    16              vout:0   cout:1  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    16              vout:0   cout:1  |
# _______________________________________________________________________________________________________________________________________
# Testing operation:      SUBU=00011   //A-B=>C    unsigned subtraction
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0010                vout:0   cout:1  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000010000    vout:0   cout:1  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    16              vout:0   cout:1  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    16              vout:0   cout:1  |
# _______________________________________________________________________________________________________________________________________
# Testing operation:       INC=00100    //A+1=>C    signed increment
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:8013                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:1000000000010011    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:-32749              vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C: 32787              vout:0   cout:0  |
# _______________________________________________________________________________________________________________________________________
# Testing operation:       DEC=00101    //A-1=>C    signed decrement
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:8011                vout:0   cout:1  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:1000000000010001    vout:0   cout:1  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:-32751              vout:0   cout:1  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C: 32785              vout:0   cout:1  |
# _______________________________________________________________________________________________________________________________________
```

ALU TB: Arithmetic Operations

```
# Testing operation:    AND=01000    //A AND B
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:8002                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:1000000000000010    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:-32766              vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C: 32770              vout:0   cout:0  |
# _______________________________________________________________________________________________________________________________________
# Testing operation:     OR =01001    //A OR B
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:8012                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:1000000000010010    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:-32750              vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C: 32786              vout:0   cout:0  |
# _______________________________________________________________________________________________________________________________________
# Testing operation:    XOR=01010    //A XOR B
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0010                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000010000    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    16              vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    16              vout:0   cout:0  |
# _______________________________________________________________________________________________________________________________________
# Testing operation:    NOT=01100    //NOT A
# ---------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:7fed                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0111111111101101    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C: 32749              vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C: 32749              vout:0   cout:0  |
# _______________________________________________________________________________________________________________________________________
```

ALU TB: Logic Operations

```
# Testing operation:       SLL=10000   //logic left shift A by the amount of B[3:0]
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0048                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000001001000    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    72              vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    72              vout:0   cout:0  |
#
# Testing operation:       SRL=10001   //logic right shift A by the amount of B[3:0]
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:2004                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0010000000000100    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:  8196              vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:  8196              vout:0   cout:0  |
#
# Testing operation:       SLA=10010   //arithmetic left shift A by the amount of B[3:0]
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0048                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000001001000    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    72              vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    72              vout:0   cout:0  |
#
# Testing operation:       SRA=10011   //arithmetic right shift A by the amount of B[3:0]
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:e004                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:1110000000000100    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C: -8188              vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C: 57348              vout:0   cout:0  |
#
```

ALU TB: Shift Operations

```
# Testing operation:       SLE=11000   //if A <= B then C(15:0) = <0...0001>
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0000                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000000000    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    0               vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    0               vout:0   cout:0  |
#
# Testing operation:       SLT=11001   //if A < B then C(15:0) = <0...0001>
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0000                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000000000    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    0               vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    0               vout:0   cout:0  |
#
# Testing operation:       SGE=11010   //if A >= B then C(15:0) = <0...0001>
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0001                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000000001    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    1               vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    1               vout:0   cout:0  |
#
# Testing operation:       SGT=11011   //if A > B then C(15:0) = <0...0001>
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0001                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000000001    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    1               vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    1               vout:0   cout:0  |
#
# Testing operation:       SEQ=11100   //if A = B then C(15:0) = <0...0001>
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0000                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000000000    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    0               vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    0               vout:0   cout:0  |
#
# Testing operation:       SNE=11101   //if A != B then C(15:0) = <0...0001>
# ---------------------------------------------------------------------------------------------------------------------------------------------------|
# hex           INPUTS: A:8012                B:8002                CarryOutEn_L:0      |   OUTPUTS: C:0001                vout:0   cout:0  |
# bin           INPUTS: A:1000000000010010    B:1000000000000010    CarryOutEn_L:0      |   OUTPUTS: C:0000000000000001    vout:0   cout:0  |
# dec(signed)   INPUTS: A:-32750              B:-32766              CarryOutEn_L:0      |   OUTPUTS: C:    1               vout:0   cout:0  |
# dec(unsigned) INPUTS: A: 32786              B: 32770              CarryOutEn_L:0      |   OUTPUTS: C:    1               vout:0   cout:0  |
```

ALU TB: Set Condition Operations

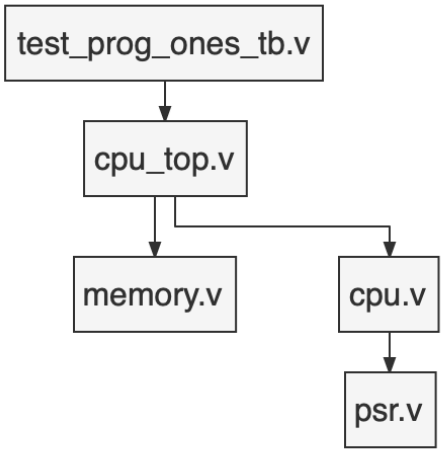
From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
B[4]	C[11]	6.309	SLOW	2.620	FAST
B[5]	C[11]	6.320	SLOW	2.620	FAST
B[6]	C[11]	6.231	SLOW	2.591	FAST
B[7]	C[11]	6.229	SLOW	2.590	FAST
A[2]	C[1]	6.076	SLOW	2.587	FAST
A[6]	C[1]	6.076	SLOW	2.587	FAST
A[10]	C[1]	6.076	SLOW	2.587	FAST
A[14]	C[1]	6.076	SLOW	2.587	FAST
B[4]	C[7]	6.347	SLOW	2.584	FAST
B[5]	C[7]	6.369	SLOW	2.583	FAST
B[4]	C[15]	6.356	SLOW	2.575	FAST
B[5]	C[15]	6.367	SLOW	2.575	FAST

ALU Critical Path: B[4] to C[11] for behavioral addition (CLA), CLA is much faster

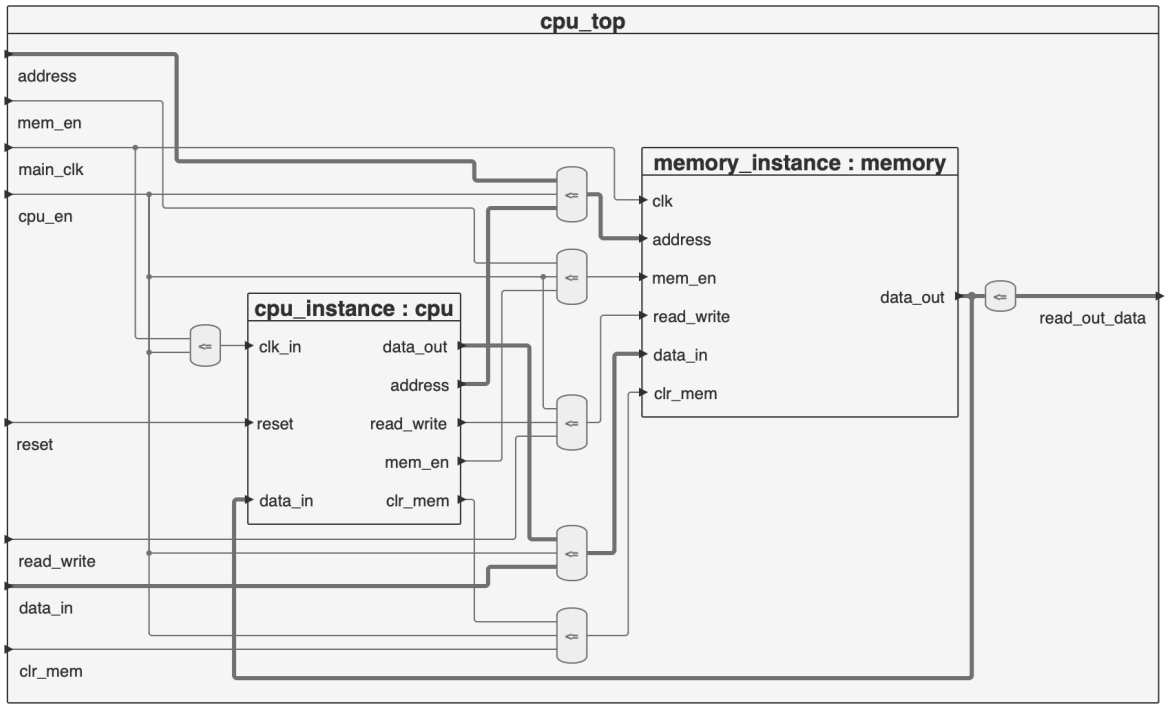
A[2]	cout	11.157	SLOW	4.641	FAST
A[2]	vout	11.136	SLOW	4.624	FAST
A[4]	cout	11.043	SLOW	4.582	FAST
A[4]	vout	11.023	SLOW	4.566	FAST
A[1]	cout	10.935	SLOW	4.528	FAST
A[1]	vout	10.915	SLOW	4.511	FAST
A[5]	cout	10.451	SLOW	4.334	FAST
A[5]	vout	10.431	SLOW	4.318	FAST
A[0]	cout	10.483	SLOW	4.302	FAST
A[0]	vout	10.463	SLOW	4.286	FAST
B[1]	cout	10.351	SLOW	4.275	FAST
B[1]	vout	10.331	SLOW	4.259	FAST
B[0]	cout	10.241	SLOW	4.205	FAST
B[0]	vout	10.221	SLOW	4.189	FAST
B[4]	C[14]	10.112	SLOW	4.163	FAST

ALU Critical Path: A[2] to cout7 for behavioral addition (ripple carry adder)

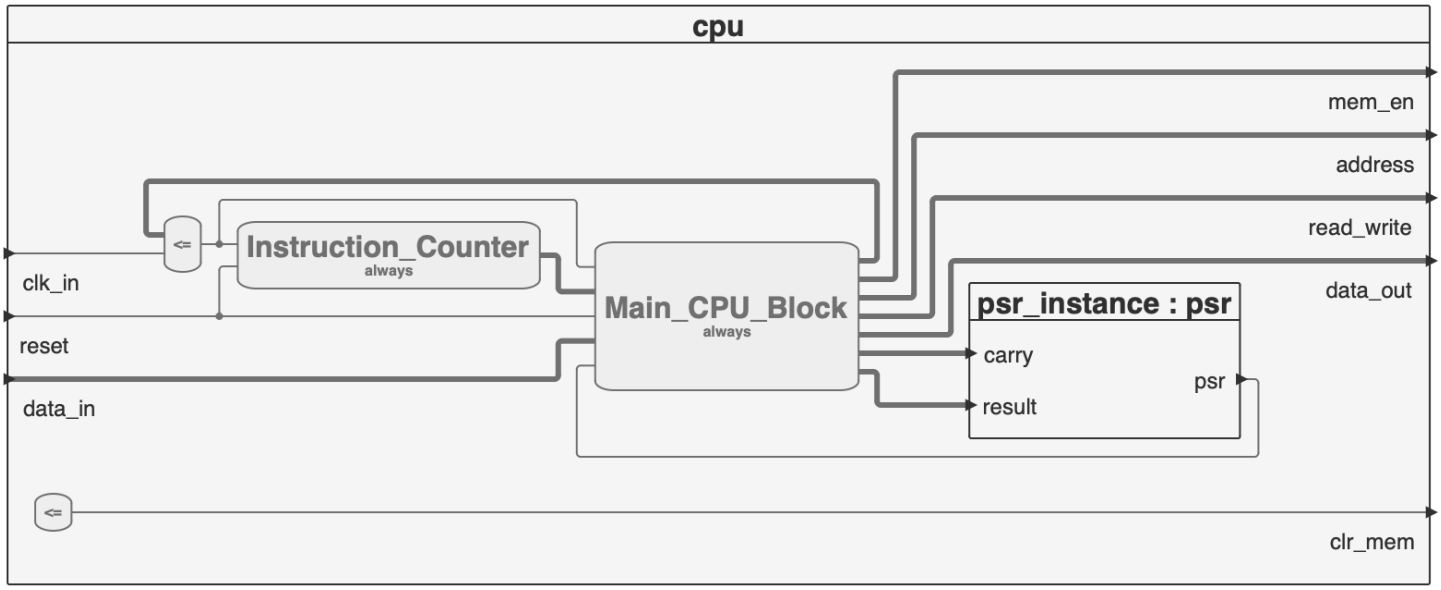
Lab Part 2_ CPU:



CPU: Verilog Module Level Architecture



CPU: Top Level Block



CPU: CPU Block

```
# ------------------------------------------------------------
# PROGRAMING MODE
# PROGRAMING N=6 TEST PROGRAM...
#   INST, SRC, DEST
#   -> LD MEM0 REG3
#   -> CMP REG3 REG4
#   -> STR REG4 MEM1
# RUNNING N=6 TEST PROGRAM...
# ****************************
# CHECKIG MEMORY VALUE AT LOCATION 1
# EXPECTED VALUE OF -6 (2s comp of 6)
# read_out_data in hex: fffffffa
# read_out_data in dec:          -6
# ------------------------------------------------------------
```

CPU: N Complement Program Testbench

Assembly Program is commented in TB above
Successful result of program is stored back in MEM and read as output of TB

Lab Part 3_ OnesCounterCPU_Program:

```
# ----------------------------------------------------------------------------
# PROGRAMING MODE
# PROGRAMING NUMBER OF ONES TEST PROGRAM...
#
# VALUES IN MEM:
# hX000  -> test_number
# hX001  -> 0
#
# PROGRAM:
# ADRS ->  INST, SRC, DEST
# hX004  -> LD MEM0 REG0
# hX005  -> LD MEM1 REG2
# hX006  -> SHF Left1 REG0
# hX007  -> BRANCH NO CARRY ADRS_11
# hX008  -> ADD IMED_1 REG2
# hX009  -> ADD IMED_0 REG0
# hX00a  -> BRANCH ZERO ADRS_12
# hX00b  -> BRANCH ALWAYS ADRS_6
# hX00c  -> STR REG2 MEM1
# hX00d  -> HLT
# RUNNING NUMBER OF ONES TEST PROGRAM...
# ************************************************************
# CHECKING MEMORY VALUE AT LOCATION 1, WHERE RESULT IS STORED
# NUMBER OF 1's IN 10101010101010100001000100010001
# ************************************************************
# read_out_data in bin: 00000000000000000000000000001100
# read_out_data in hex: 0000000c
# read_out_data in dec:          12
# ----------------------------------------------------------------------------
```

CPU: Ones Counter Program Testbench

Assembly Program is commented in TB above

Successful result of program is stored back in MEM and read as output of TB

Lab Part 4_ MultiplierCPU_Program:

```
# ----------------------------------------------------------------------
# PROGRAMING MODE
# PROGRAMING MULTIPLICATION TEST PROGRAM...
#
# VALUES IN MEM:
# hX000  -> A VALUE = 0000000e
# hX001  -> B VALUE = 0000000f
# hX002  -> C INITIAL = 00000000
# hX00f  -> DECREMENT = ffffffff
#
# ADRS  -> INST SRC DEST
# hX004 -> LD MEM0 REG0
# hX005 -> LD MEM1 REG1
# hX006 -> LD MEM2 REG2
# hX007 -> LD MEMf REGf
# hX008 -> ADD REG0 REG2
# hX009 -> ADD REGf REG1
# hX00a -> BRA ZERO ADRS00c
# hX00b -> BRA POS ADRS008
# hX00c -> STR REG2 MEM2
# hX00d -> HLT
# RUNNING MULTIPLICATION TEST PROGRAM...
# ****************************
# CHECKING MEMORY VALUE AT LOCATION F3
# MULTIPLIED 14 * 15, EXP RES: 210
# read_out_data in bin: 00000000000000000000000011010010
# read_out_data in hex:             000000d2
# read_out_data in dec:                  210
# ----------------------------------------------------------------------
```

CPU: Multiplication Program Testbench

Assembly Program is commented in TB above
Successful result of program is stored back in MEM and read as output of TB