LAB 3

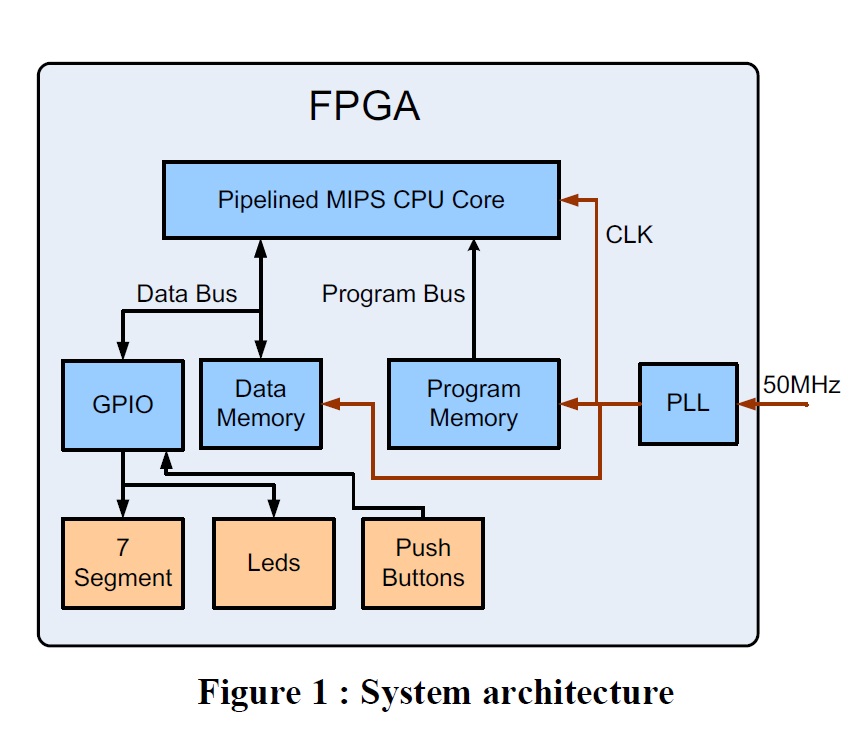
Dor Livne 204528251

Amir Tsur 203795828



Full System:

The diagram illustrating the system is

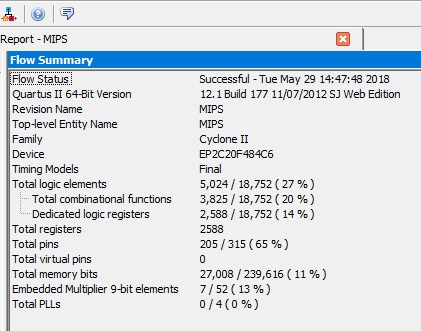


*Port table:*

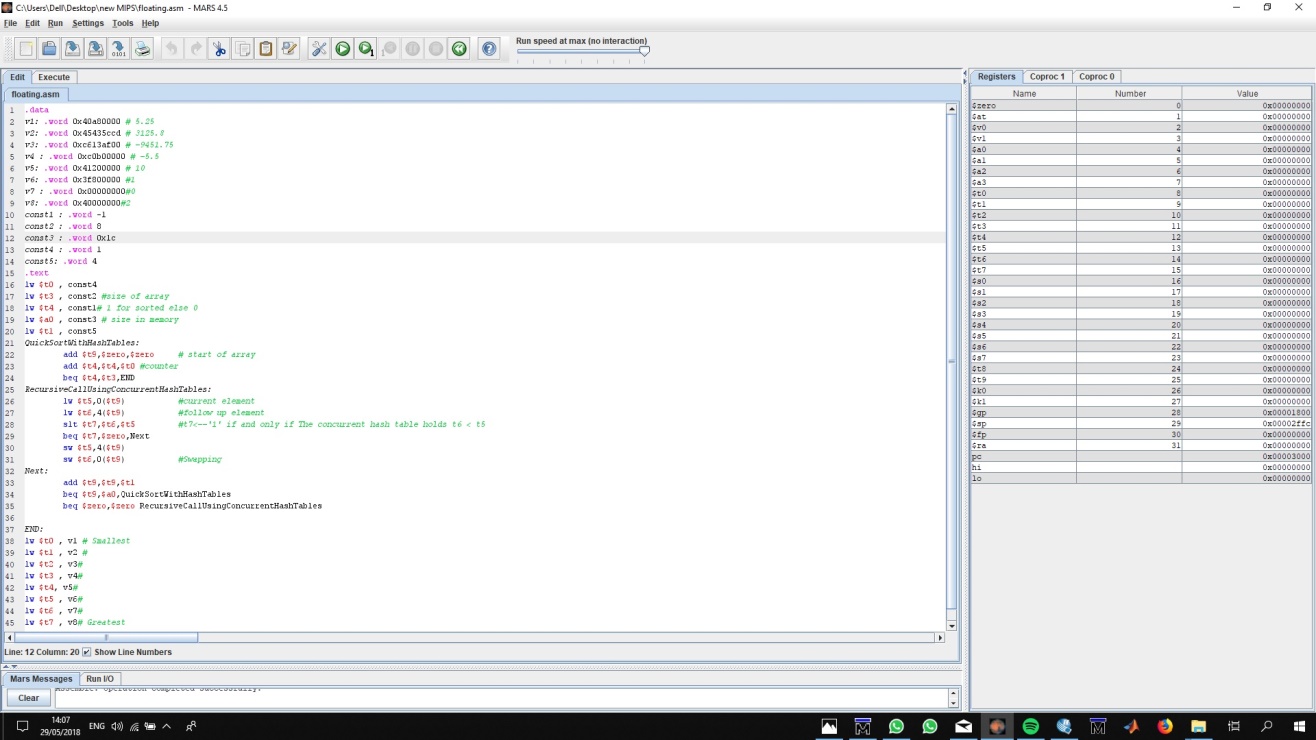
|  |  |  |  |
| --- | --- | --- | --- |
| Port | direction | size | functionality |
| Sw8 | in | 1 bit | Hex(4 downto 1)<-FPU\_Hi if 1 else FPU\_LO |
| clk | in | 1 bit | Clk<-Clock\_50 |
| Status | Out | 6 bit | LEDG<-Status |
| Hex1,Hex2,Hex3,Hex4 | Out | 4 bit each | 7\_seg<-Hex |

The full system design is fed from predetermined program, written in assembly using MARS.

**Logic\_elements usage:**

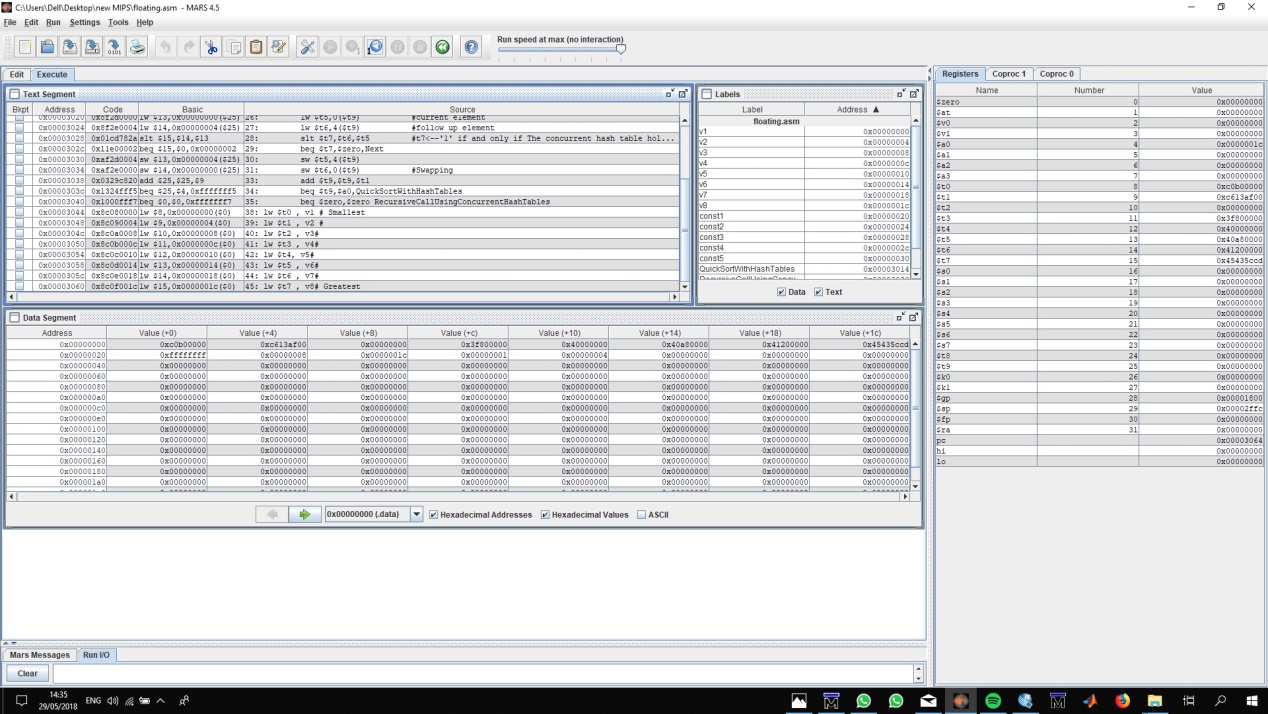


**Mars simulation**

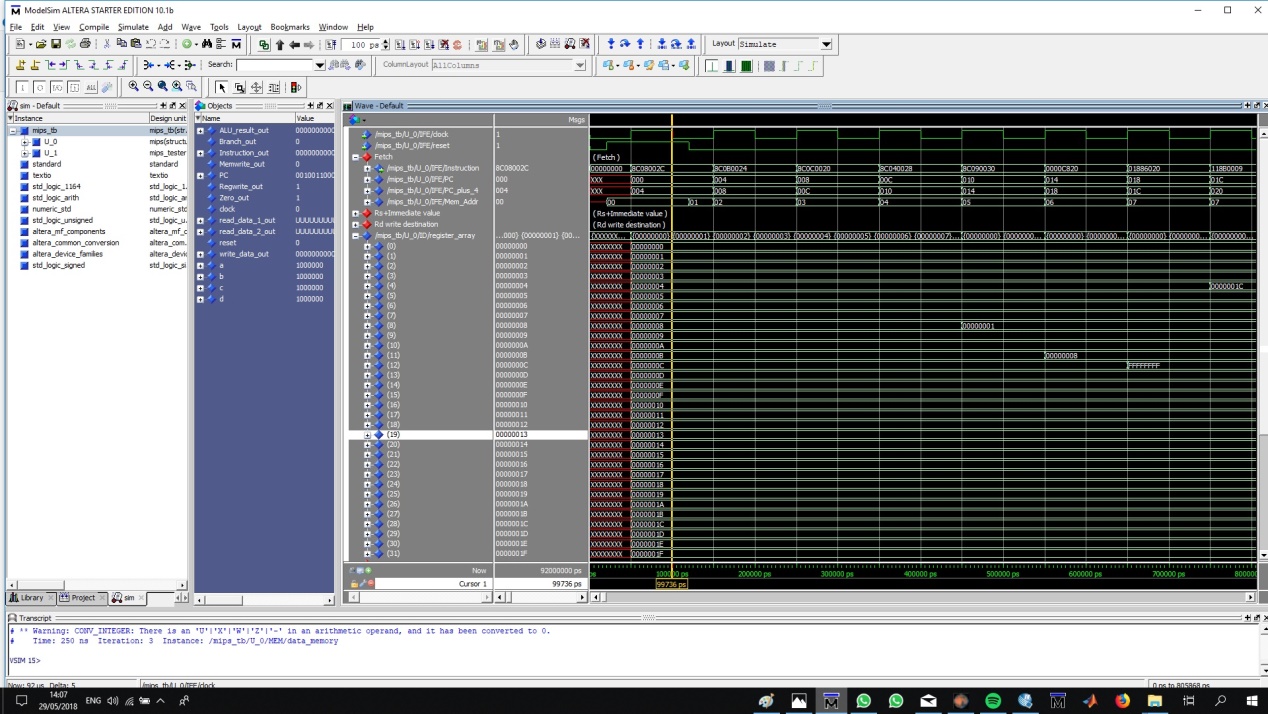


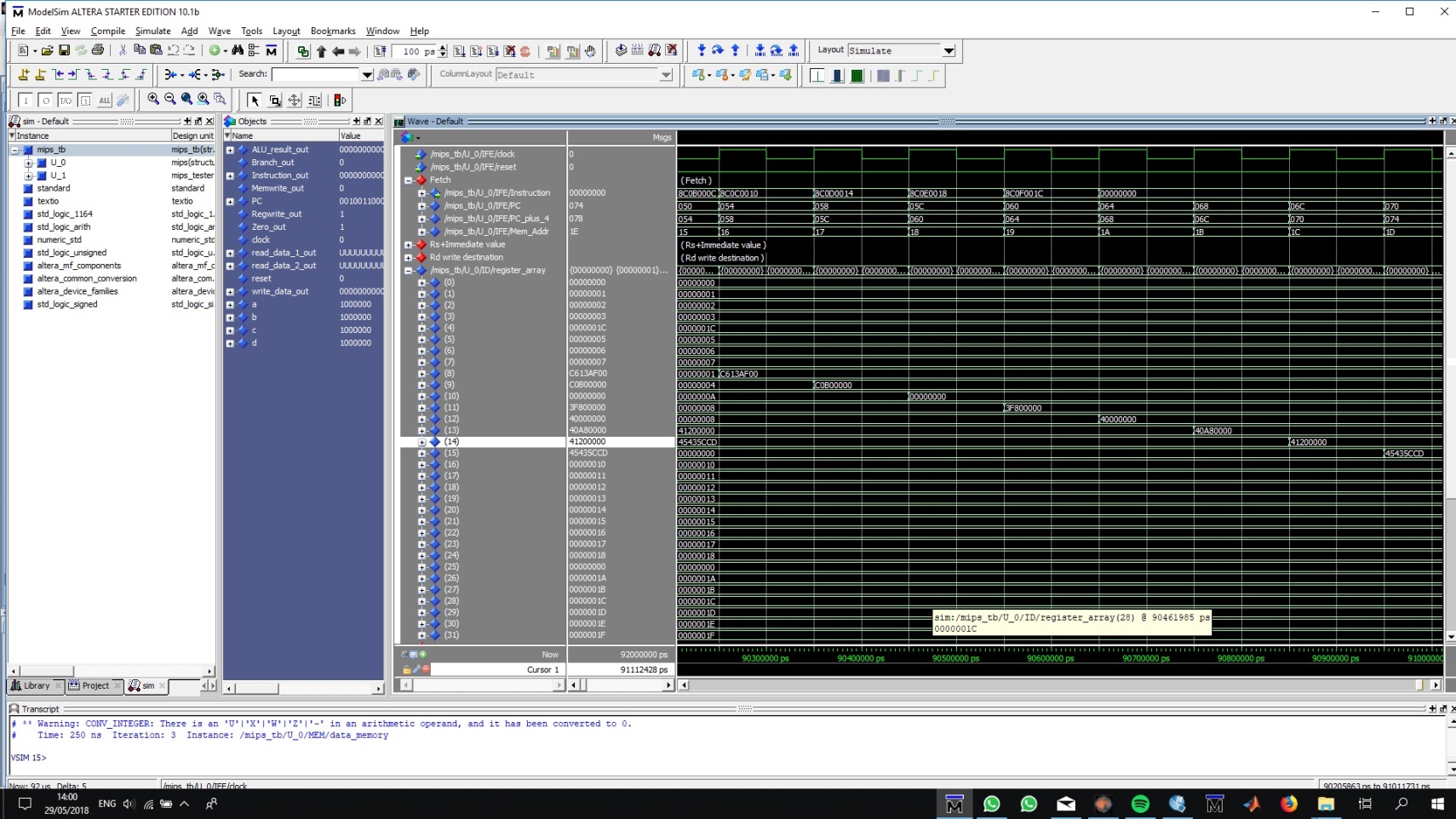
The code and the initial value of the registers by MARS

**Simulation by mars:**

****

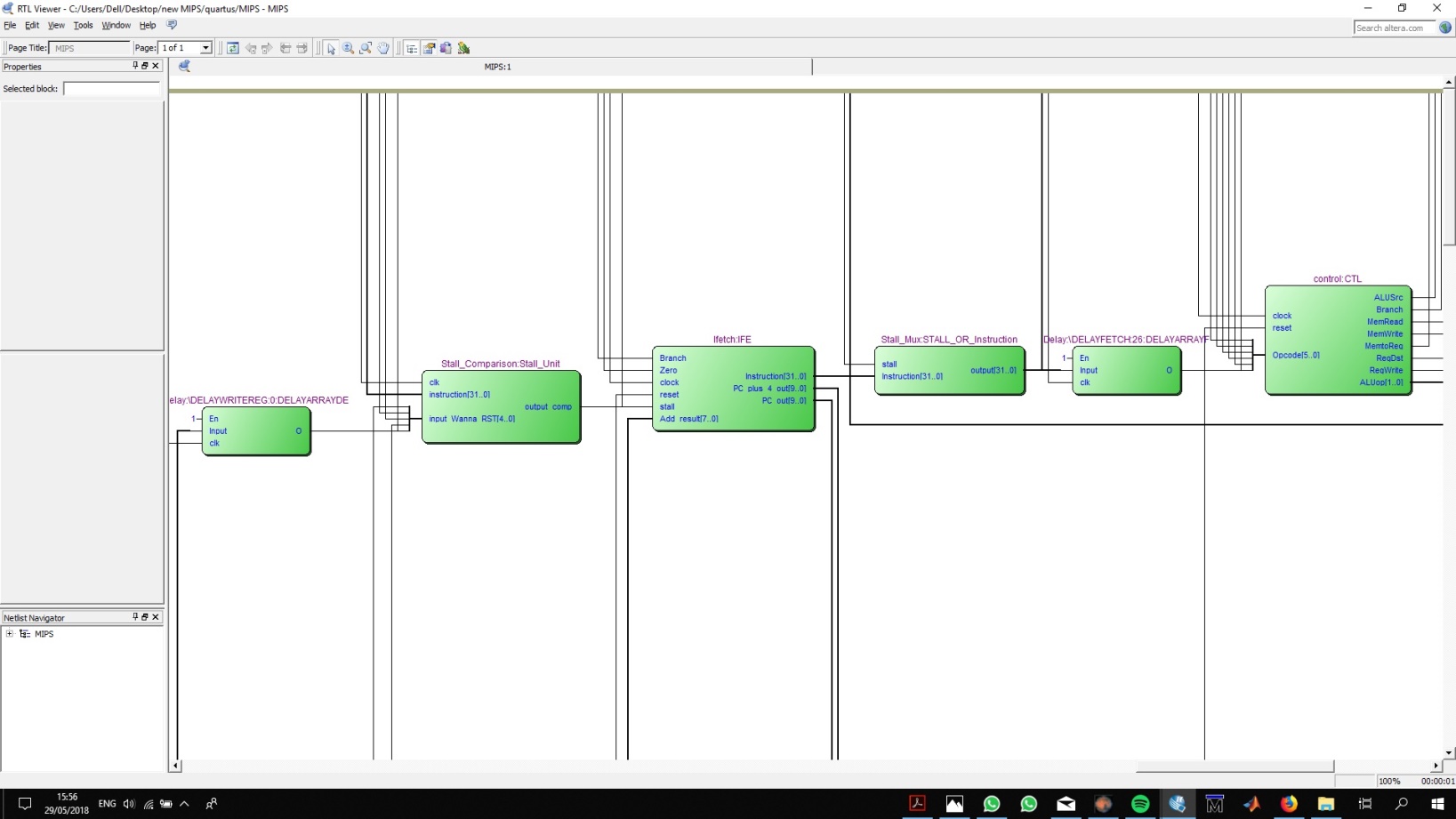
**Initial values by ModelSim**

****

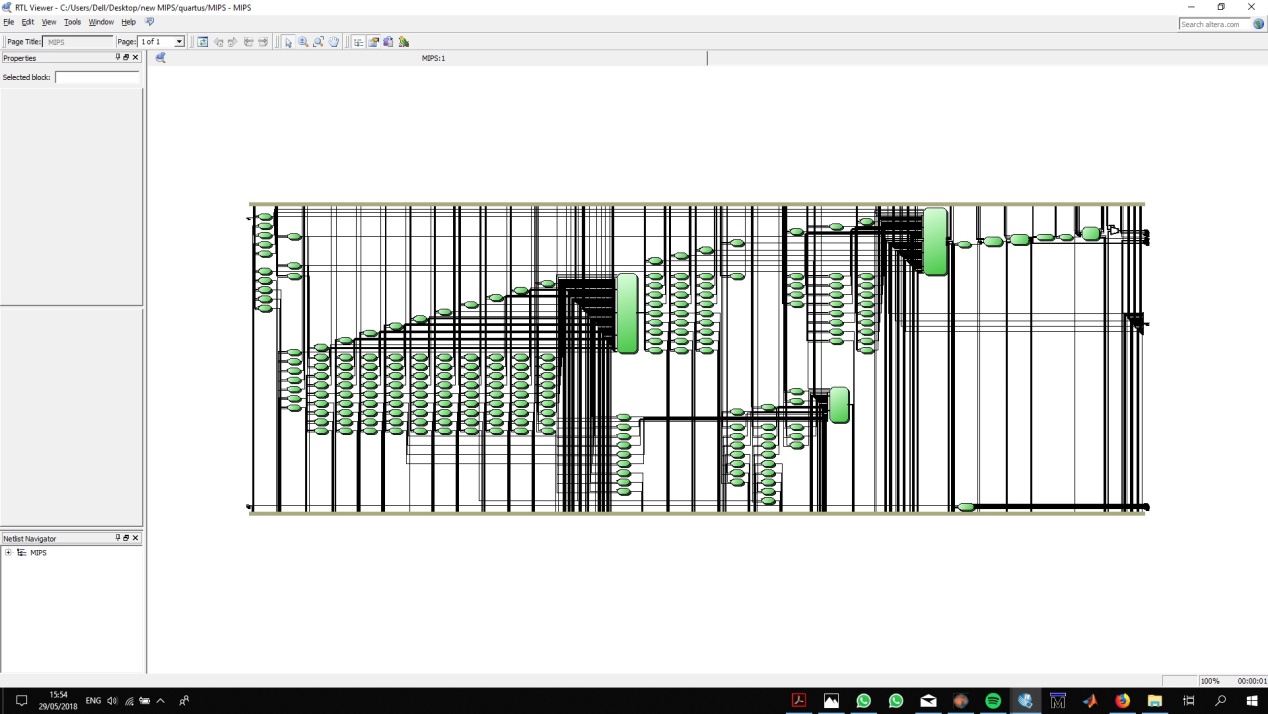
**Output – sorted vector by ModelSim**

**The result is the same to mars simulation.**

**RTL\_view:**

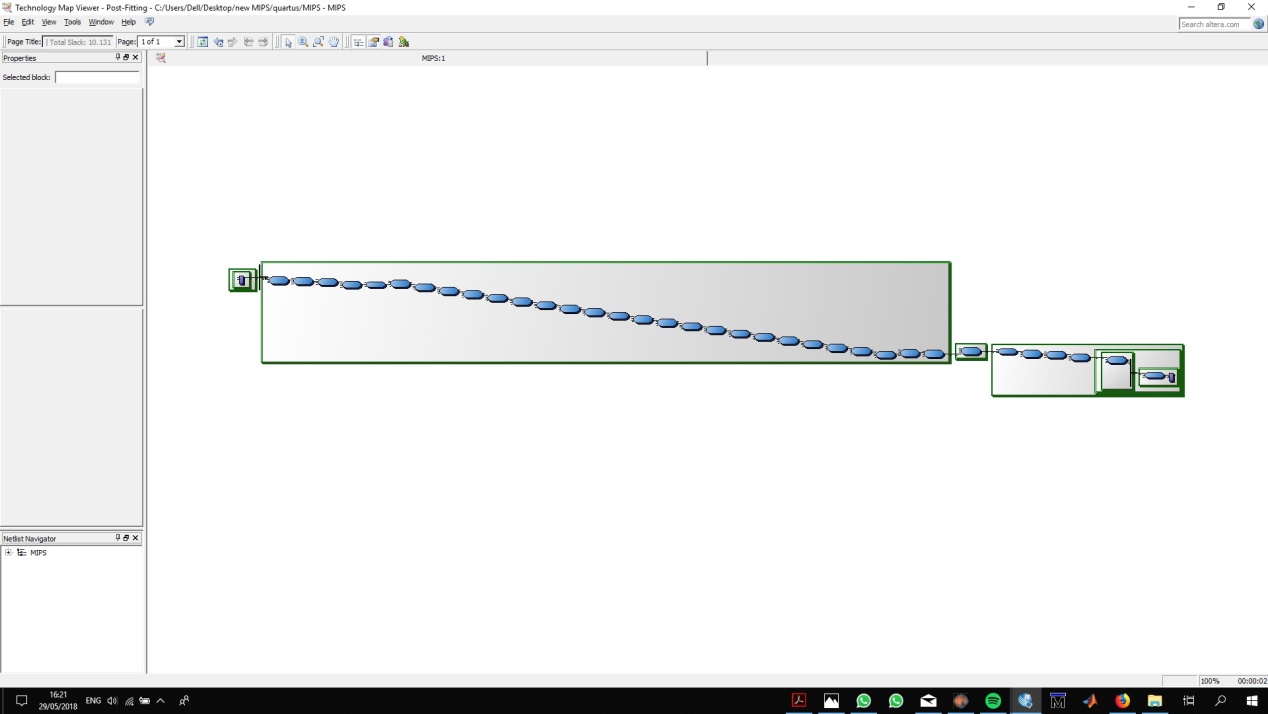


Hazard Unit



*Critical Path:*

Our Fmax is 15.1MHZ (slow model).

**

The critical path consists of the floating point system.

*HAZRAD UNIT*

This unit detects and deals with 3 types of hazards: (source – Pipeline lectures)

1. *Data Hazards –* An Instruction in flight wants to use a data value that’s not “done” yet – Instruction depends on result of prior computation which is not ready yet.
2. *Structural Hazards-* Hardware can’t support a combination of instructions.
3. *Control Hazards –* Pipeline of branches and other instructions which change the PC.

Measures taken to deal with each hazard:

1. Stall.
2. Double Pump.
3. Branch Detection and Computation at Fetch Phase.
4. Redesign of resources – alternation of several command to behave differently (XOR, NOR and SLT).

Regular cases that were checked:

1. Instructions without hazards
2. Dependency of branches (taken/not taken) and the following instructions
3. Usage of the redesigned of resources.

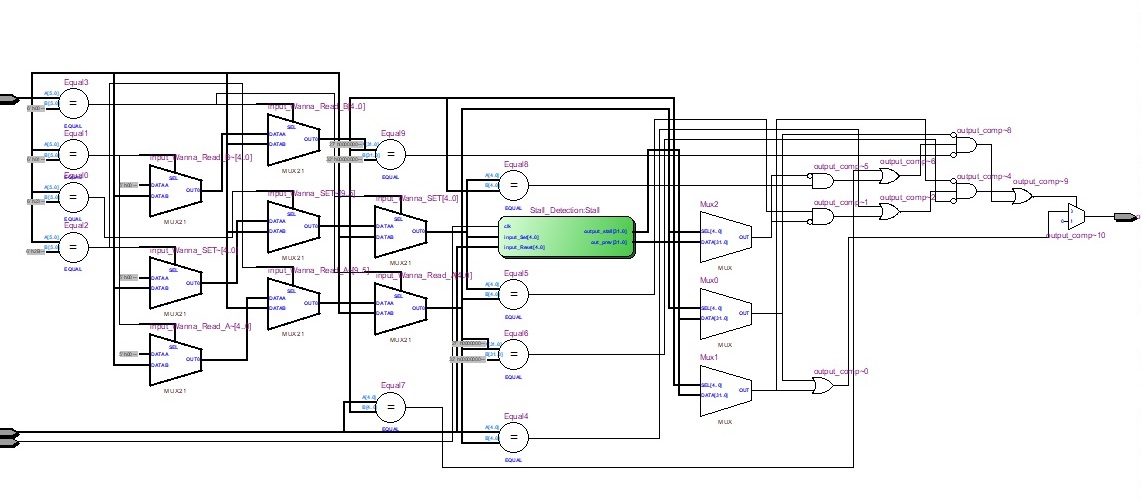
Extreme cases that were checked:

1. Reading from the same register we want to write to – no hazard, the system should continue in fetching (eg. Add $t1,$t1,$t1).
2. Resetting and Acquiring the same register in the same clock – (e.g add $t1(restes),$t1,$t1, clock, clock, add $t1(acquire),$t1,$t1)

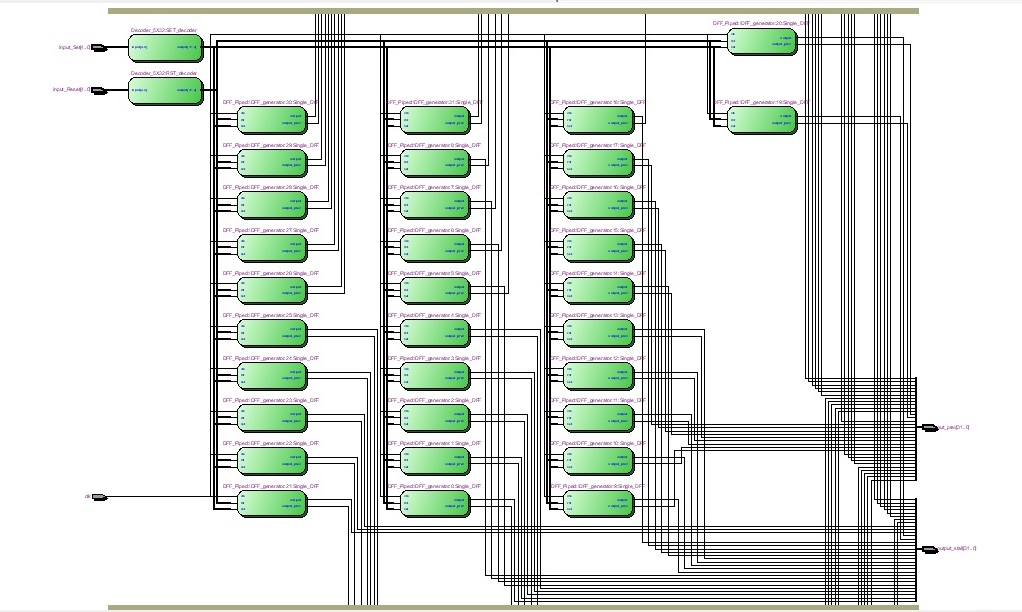
Forbidden cases that were not checked:

1. Support of new intsructions (addi,lui, etc).
2. Support of comparison and usage with scalars, instead if a premade register.

*Hazard Unit*



**Stall-unit**



|  |  |  |  |
| --- | --- | --- | --- |
| Port | direction | size | functionality |
| clk | In | 1 | clock |
| Wanna\_Reset | In | 32 |  |
| Instruction | in | 32 |  |
| Output\_cpmp | out | 1 | Out🡨hazard/safe |

This unit gets the current instruction and analyzes its components:

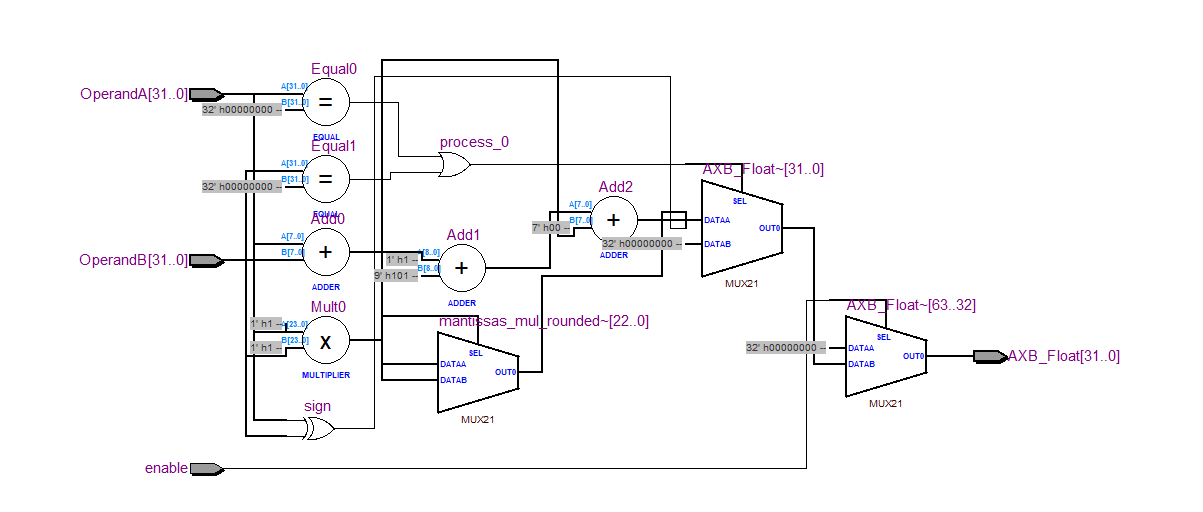
1. Opcode type – R-format,I-Type,J-Type
2. Registers needed to read from – depends on the opcode type – each type handled differently.

The unit also gets the register now cleared (Wanna\_Reset) – when we start the WB phase.

The unit also stores list of occupied/available registers for current time and the clock beforehand (by the equation )

The unit then decides whether we are good to go or we detected a hazard.

*FPM:*



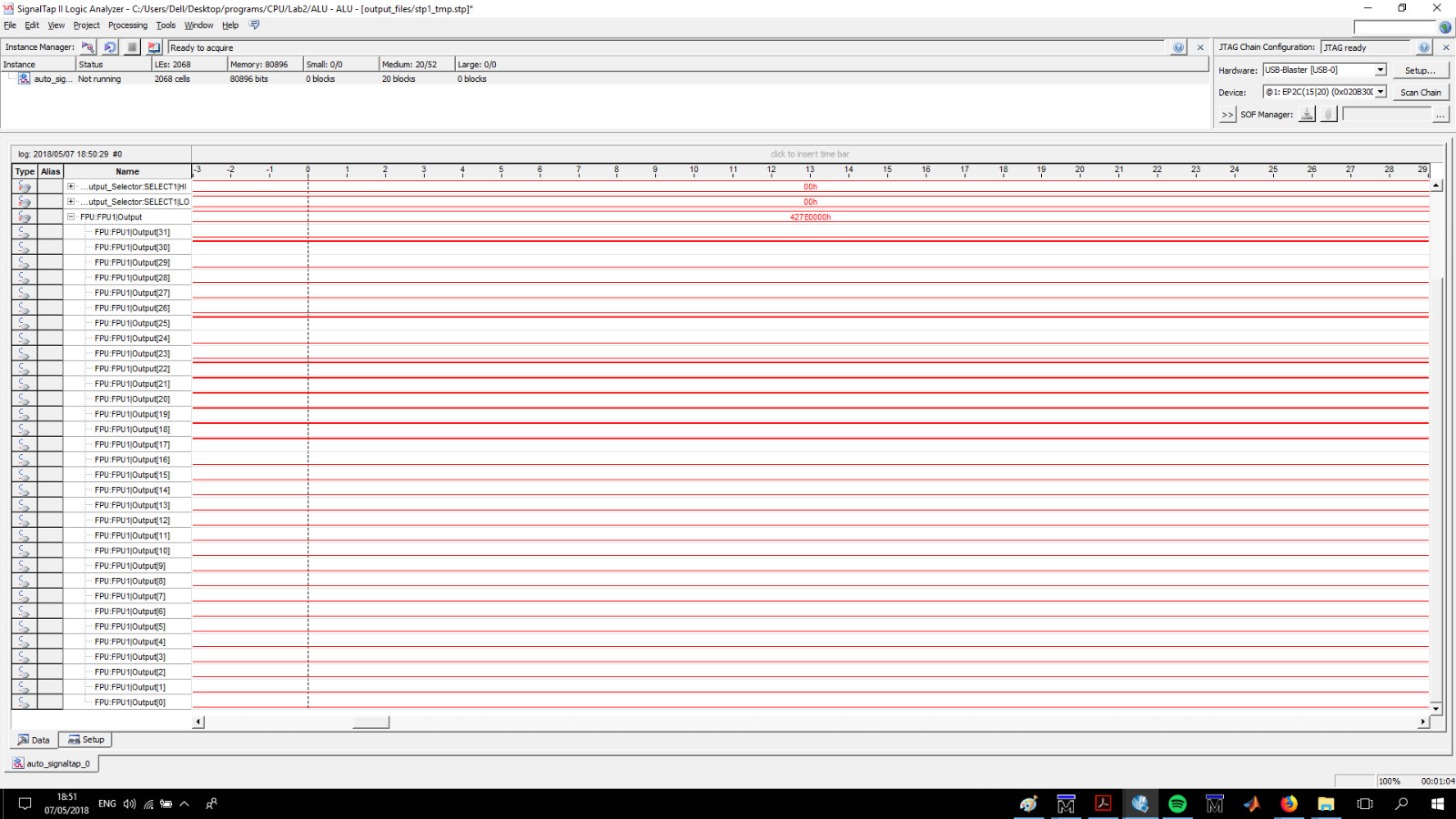
|  |  |  |  |
| --- | --- | --- | --- |
| Port | direction | size | functionality |
| OperandA | In | 32 | Floating point IEEE |
| OperandB | In | 32 | Floating point IEEE |
| enable | In | 1 |  |
| AxB\_Float | out | 32 | Out🡨A\*B |

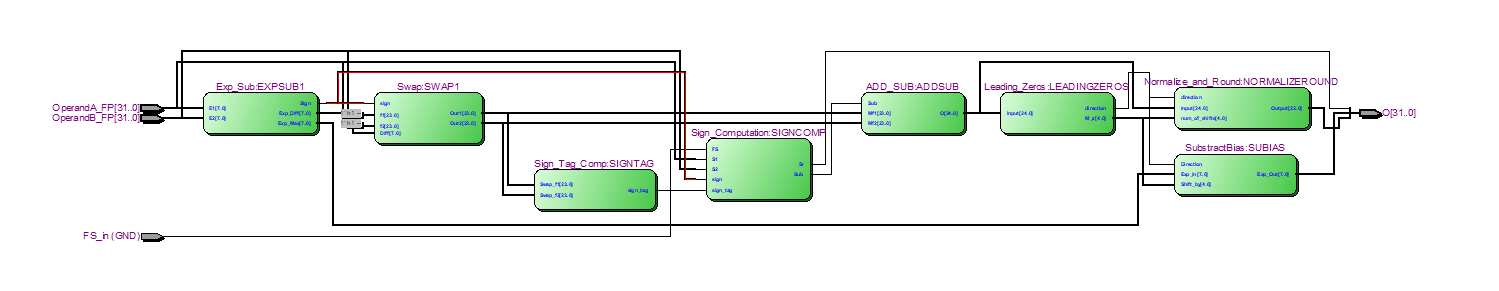
This Unit is responsible for multiplying two floats, it seperates the two floats to

Two mantissas and two exponents

The two manitssas are multiplied while the two exponents are added together

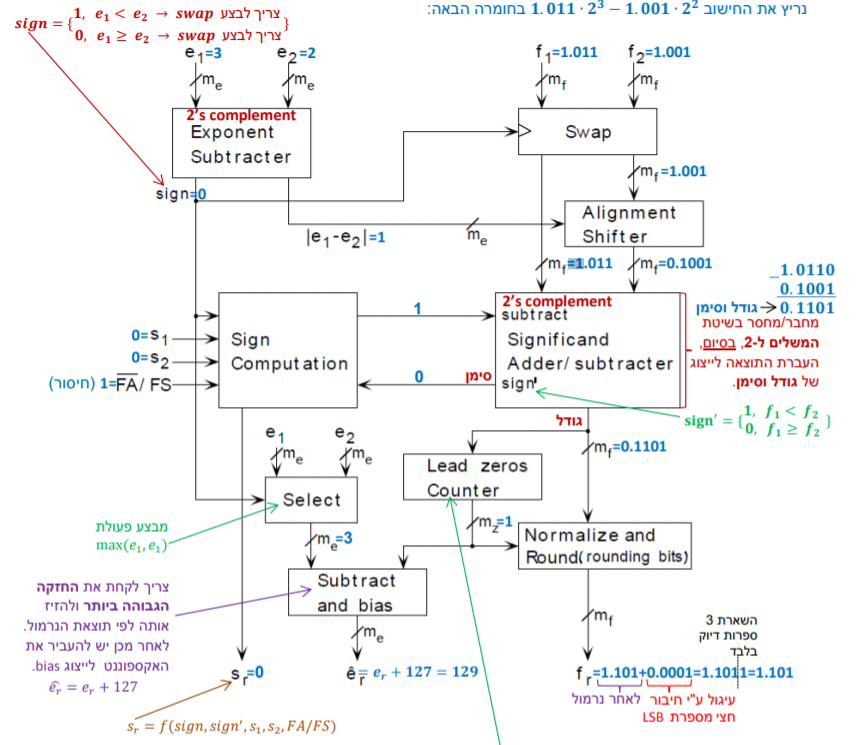
Eventually we put the result in AXB\_Float and transmit it out of this module.

FPM SIGNAL TAP

*FPA*

|  |  |  |  |
| --- | --- | --- | --- |
| Port | direction | size | functionality |
| OperandA | In | 32 | Floating point IEEE |
| OperandB | In | 32 | Floating point IEEE |
| Fs\_in | in | 1 | Fs='1'🡪A-B,Fs='0'🡪A+B |
| O | out | 32 | Out🡨A+/-B |

OperandA and OperandB are two floats and O is A+B or A-B according to the value of Fs. This unit implements the next diagram :



***Quick explanation of the units (same to previous assignment):***

**Exp\_Sub:**

the input to this unit is the E1 – Exponent of A, E2-Exponent of B

The output of this system is Diff ,sign and Exp\_Max where :

**Swap:**

the input of this unit is F1- mantissa of A normalized which is 23 bit,

F2- mantissa of B normalized which is 23 bit,

sign-'1'-> switch,

Difff-> how much to shift left Swap\_out2

The output of this unit is

Out1 and Out2 and they are the inputs to ADD\_SUB

**Sign\_Tag\_Computation:**

the input of this unit is Out1,Out2

The output is sign' which is determine by:

**ADD\_SUB**:

the inputs of this unit is Out1,Out2 and FS .

the output is O which is of size 24 bit

**Leading\_Zeros:**

Input is O which is 24 bits

Output is M\_z and direction which M\_z is the how much we need to shift the O inorder to make in normalized and direction is to which side to shift.

**Normalize and Round:**

Input is O ,direction,M\_z

Output is *Output* which is O normalized and which is 22 bit(Rounded O )

**Sign\_Computation:**

Input is sign,S1->the sign of A ,S2->the sign of B,Fs,Sign'

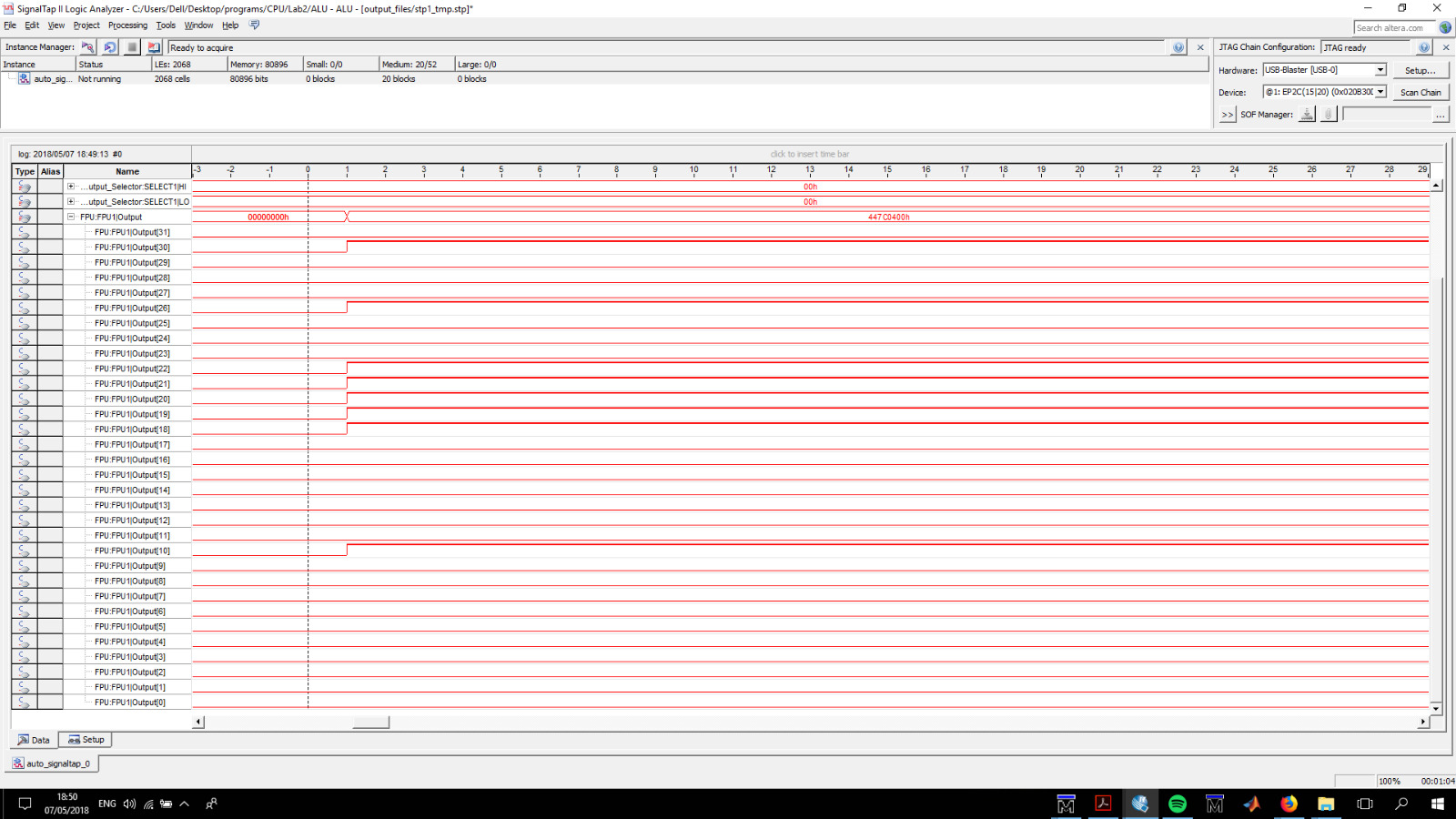
Output is Sr-> the sign of the output and Subtract-> this is the FS\_in of ADD\_SUB Unit.

**SubtractBias:**

Input is Exp\_Max ,direction and M\_z as described above

Output is Exp\_Out which is 7 bit and is described as

FPA SIGNAL TAP



**Conclusions and further improvements**

We believe that we managed to accomplish this lab's targets.

Always there is a room for improvement – given the time to do so.

We believe we could accelerate the system’s CPI and it’s Period time (Fmax) with several modifications:

1. Deepen the Pipeline’s depth – break the Instruction computation to more phases – shorter operations🡪Fmax increases.
2. Support I-type instructions – less stages in application layer.
3. Make floating point computation compatible with the pipeline – this is the critical path and break it into stages will make a great performance improvement.