VLSI LAB 1 EE513

Experiment No. 09

Pawan Kumar
(214102412)
EEE department, Indian Institute of Technology Guwahati

13th november 2021

Design an HDL-based 32-bit processor (Instruction decode (Instruction decoder + controller) + Instruction execute (ALU)+register write (Register Bank)) which executes the following 10 instructions:

ADD SUB SLL SLT SLTU XOR SRL SRA OR AND

Objective:

Perform post-synthesis simulations for a basic 32-bit processor which executes the aforementioned instructions. The processor will basically have the following modules:

- A 32-bit register bank which will have 32 registers and each register can store 32-bit data.
- An instruction decoder module which will slice the 32-bit instructions into the corresponding fields opcode, function and rs1, rs2 and rd values.
- A controller module which will generate a corresponding signal for the respective operations depending function and opcode.
- 32-bit ALU for processing the data present in rs1 and rs2 registers.
- Register write to store the result back into rd register in the register bank.

Tool Used:

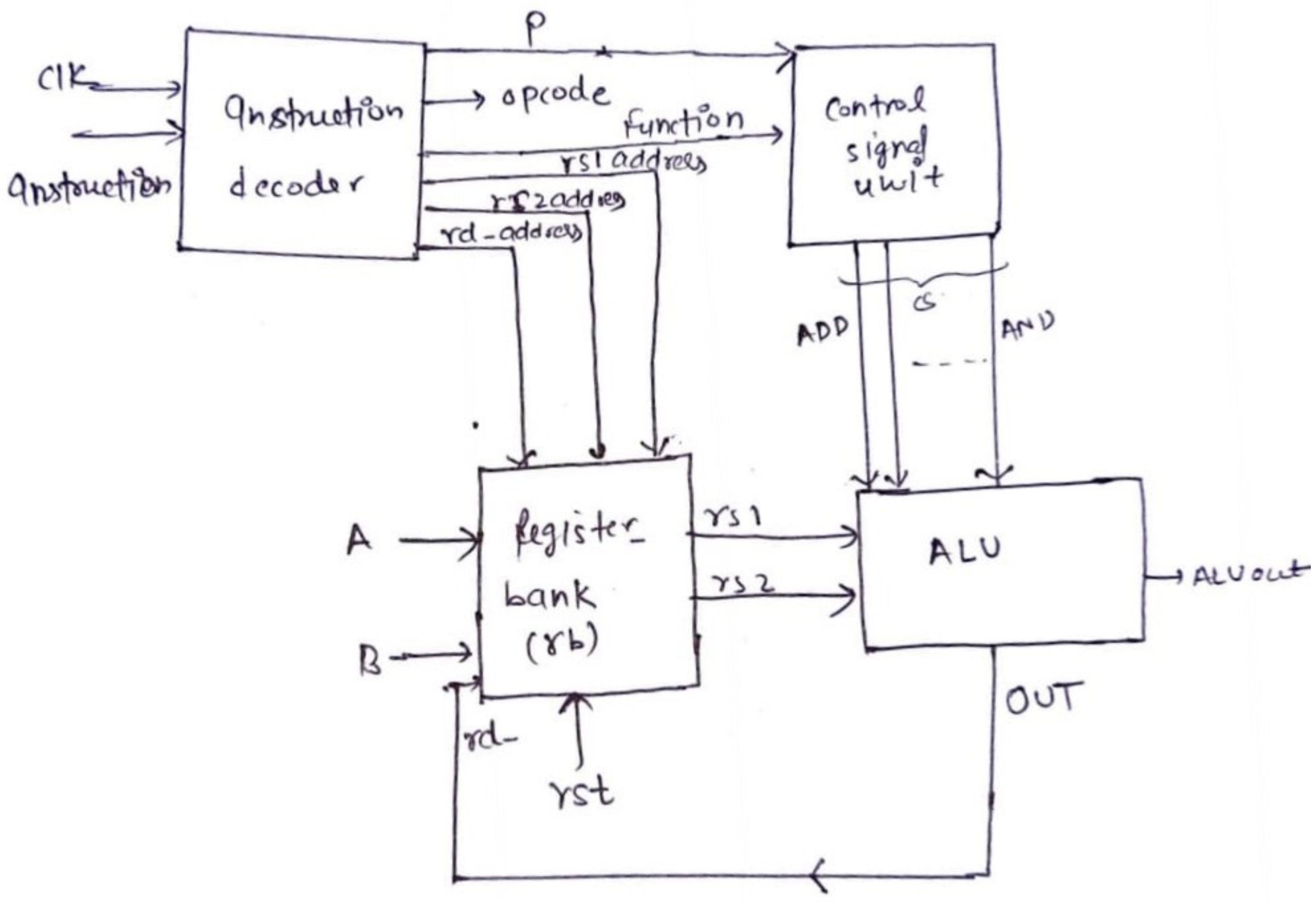
Xilinx 2019.1

Theory:

32 bit processor can process 32 bit Instructions per clock cycle. It operates the given function (In our case the functions are mentioned in above page) between two 32 bit data (which is called operands) and provides result of 32 bit too.

In more technical terms, this means processors can work with 32-bit binary numbers (decimal number up to 4,294,967,295). Anything larger and the computer would need to break the data into smaller pieces.

Each instruction have opcode of 7 bit(0 to 6),function of 3 bit(12 to 14),address of 1st and 2nd operands (rs1 (bit 15-19), rs2 (bit 20-24)) and Address of result also of 5 bit (rd (bit 7-11)). 31th bit of instruction for differentiate between ADD and SUB, SRL and SRA.



Basic block diagram of 32 bit processor

Instruction Decoder:

When we provide the 32 bit instruction to processor this decoder decodes the 7-bit opcode, 5 bit address of Register (rs1,rs2 and rd) , 3 bit function.

- Opcode 7 bit (0-6)
- Register addresses (for rs1, rs2 and rd) of 5 bit
- Function 3 bit
- p is 1 bit (31th bit)

Register bank:

Basically register bank (32X32) is capable of storing 32 bit data at each location.The operands of 32 bit data is written at the address of operands(rs1 and rs2) and the result will be overwrite at the address of rd.

Control signal unit:

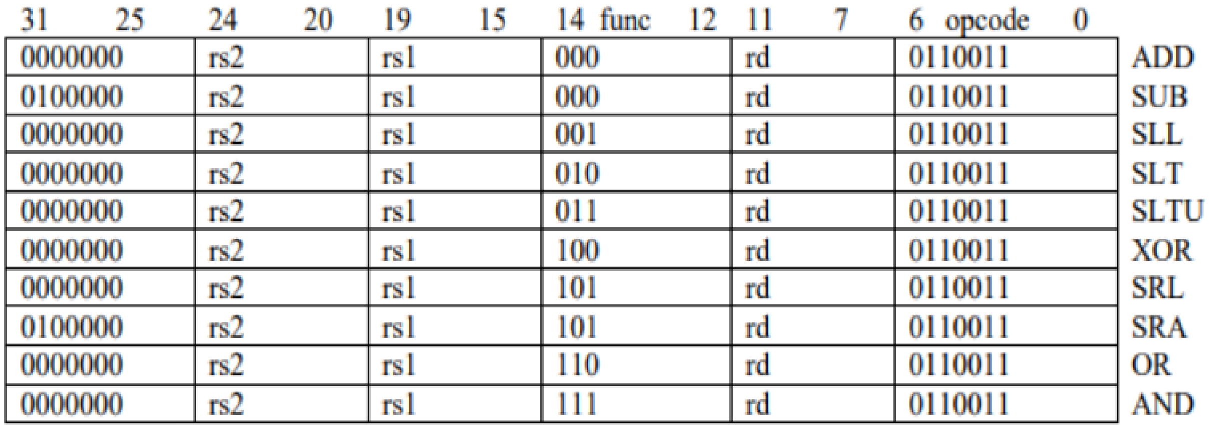
It will take the input as function and bit p and accordingly output will be the required function as (ADD SUB SLL SLT SLTU XOR SRL SRA OR AND).

- Out cs is of 4 bit which tells the operator.

ALU:

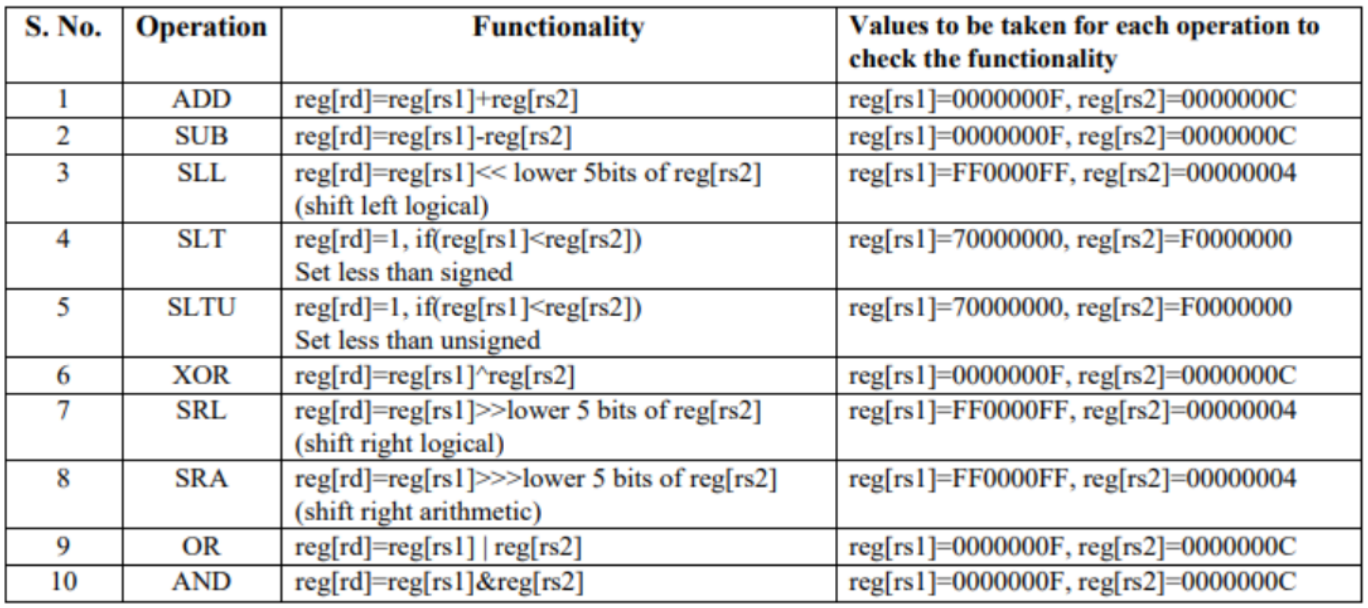
The main functionality of this unit is to performing the given operation between two 32 bit data and providing the result which will be stored in given address 32 bit register.

Instruction set Specification of the 32-bit processor:



31 25	24 20	19 15	14 func 12	11 7	6 opcode 0	
0000000	rs2	rs1	000	rd	0110011	ADD
0100000	rs2	rs1	000	rd	0110011	SUB
0000000	rs2	rs1	001	rd	0110011	SLL
0000000	rs2	rs1	010	rd	0110011	SLT
0000000	rs2	rs1	011	rd	0110011	SLTU
0000000	rs2	rs1	100	rd	0110011	XOR
0000000	rs2	rs1	101	rd	0110011	SRL
0100000	rs2	rs1	101	rd	0110011	SRA
0000000	rs2	rs1	110	rd	0110011	OR
0000000	rs2	rs1	111	rd	0110011	AND

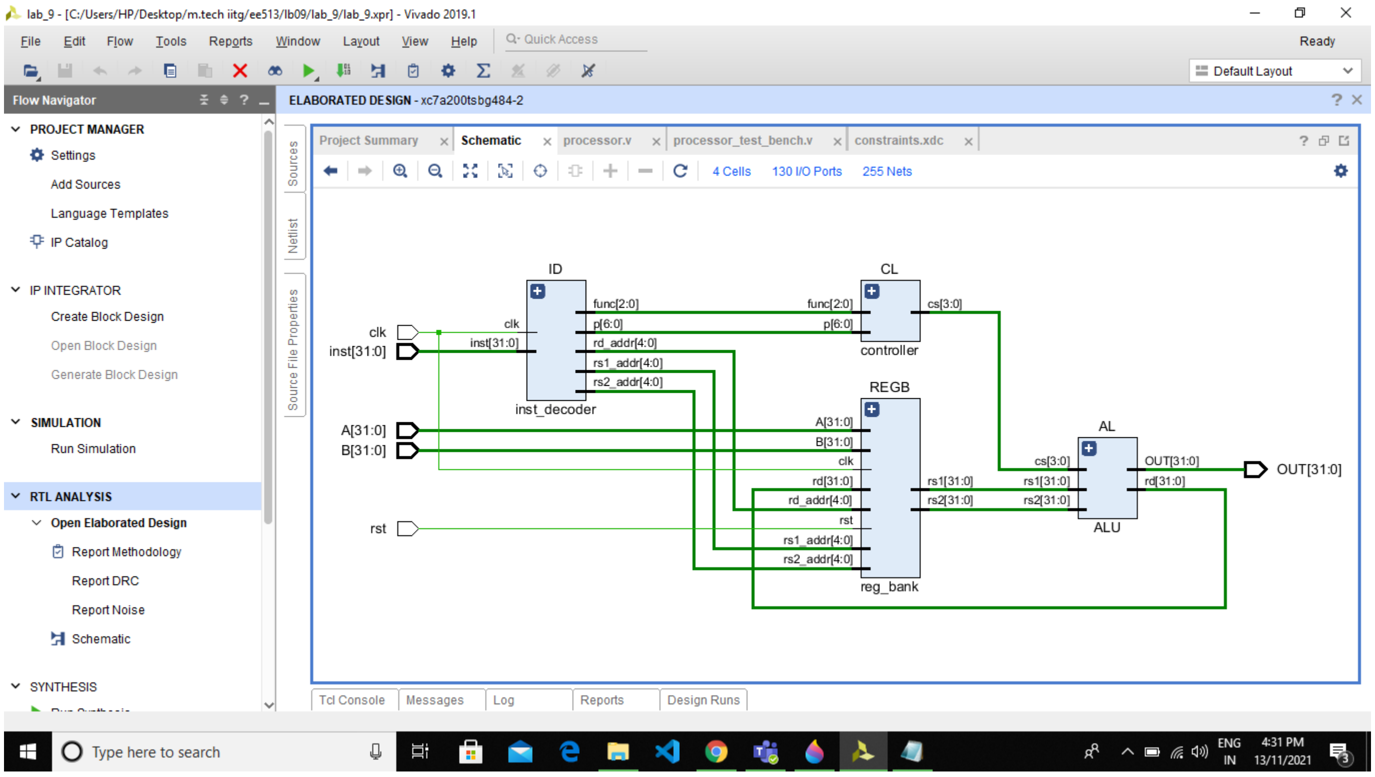
Operations performed by ALU for each instruction:



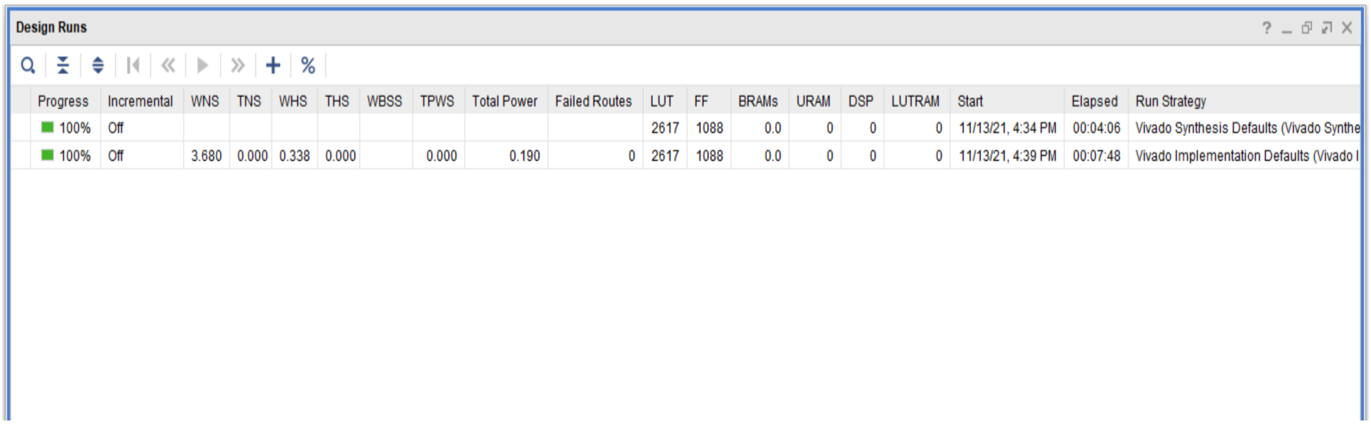
S. No.	Operation	Functionality	Values to be taken for each operation to check the functionality
1	ADD	reg[rd]=reg[rs1]+reg[rs2]	reg[rs1]=0000000F, reg[rs2]=0000000C
2	SUB	reg[rd]=reg[rs1]-reg[rs2]	reg[rs1]=0000000F, reg[rs2]=0000000C
3	SLL	reg[rd]=reg[rs1]<< lower 5bits of reg[rs2] (shift left logical)	reg[rs1]=FF0000FF, reg[rs2]=00000004
4	SLT	reg[rd]=1, if(reg[rs1]<reg[rs2]) Set less than signed	reg[rs1]=70000000, reg[rs2]=F0000000
5	SLTU	reg[rd]=1, if(reg[rs1]<reg[rs2]) Set less than unsigned	reg[rs1]=70000000, reg[rs2]=F0000000
6	XOR	reg[rd]=reg[rs1]^reg[rs2]	reg[rs1]=0000000F, reg[rs2]=0000000C
7	SRL	reg[rd]=reg[rs1]>>lower 5 bits of reg[rs2] (shift right logical)	reg[rs1]=FF0000FF, reg[rs2]=00000004
8	SRA	reg[rd]=reg[rs1]>>>lower 5 bits of reg[rs2] (shift right arithmetic)	reg[rs1]=FF0000FF, reg[rs2]=00000004
9	OR	reg[rd]=reg[rs1] | reg[rs2]	reg[rs1]=0000000F, reg[rs2]=0000000C
10	AND	reg[rd]=reg[rs1]&reg[rs2]	reg[rs1]=0000000F, reg[rs2]=0000000C

Observations:

Schematic of the processor:



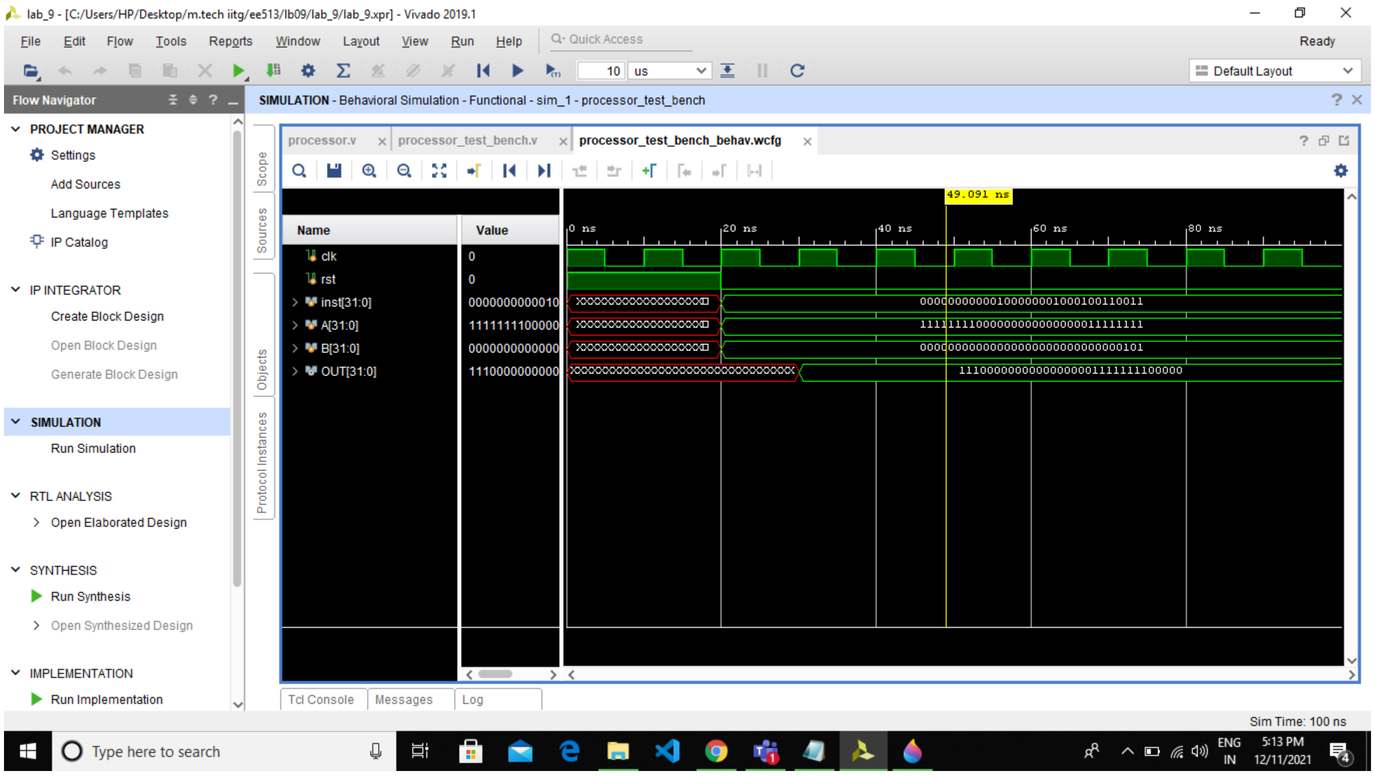
No of LUTs and Flip Flops used in the processor:


Design Runs

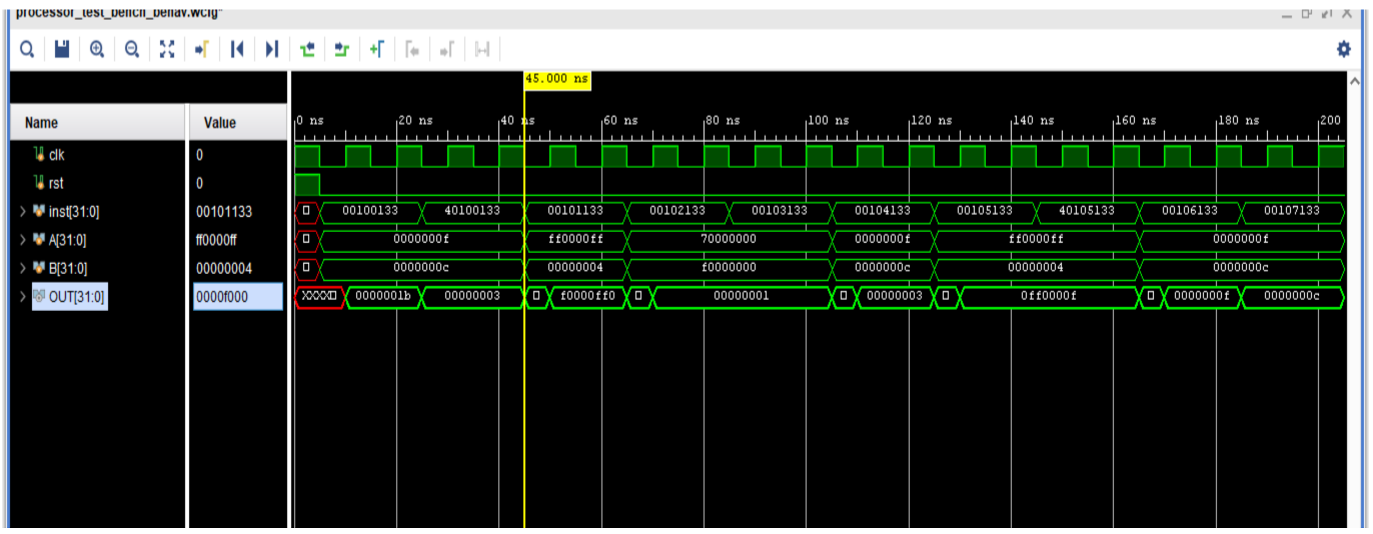
Progress	Incremental	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	LUTRAM	Start	Elapsed	Run Strategy
🟩 100%	Off									2617	1088	0.0	0	0	0	11/13/21, 4:34 PM	00:04:06	Vivado Synthesis Defaults (Vivado Synthe
🟩 100%	Off	3.680	0.000	0.338	0.000		0.000	0.190	0	2617	1088	0.0	0	0	0	11/13/21, 4:39 PM	00:07:48	Vivado Implementation Defaults (Vivado I

Output Wave form of the processor:

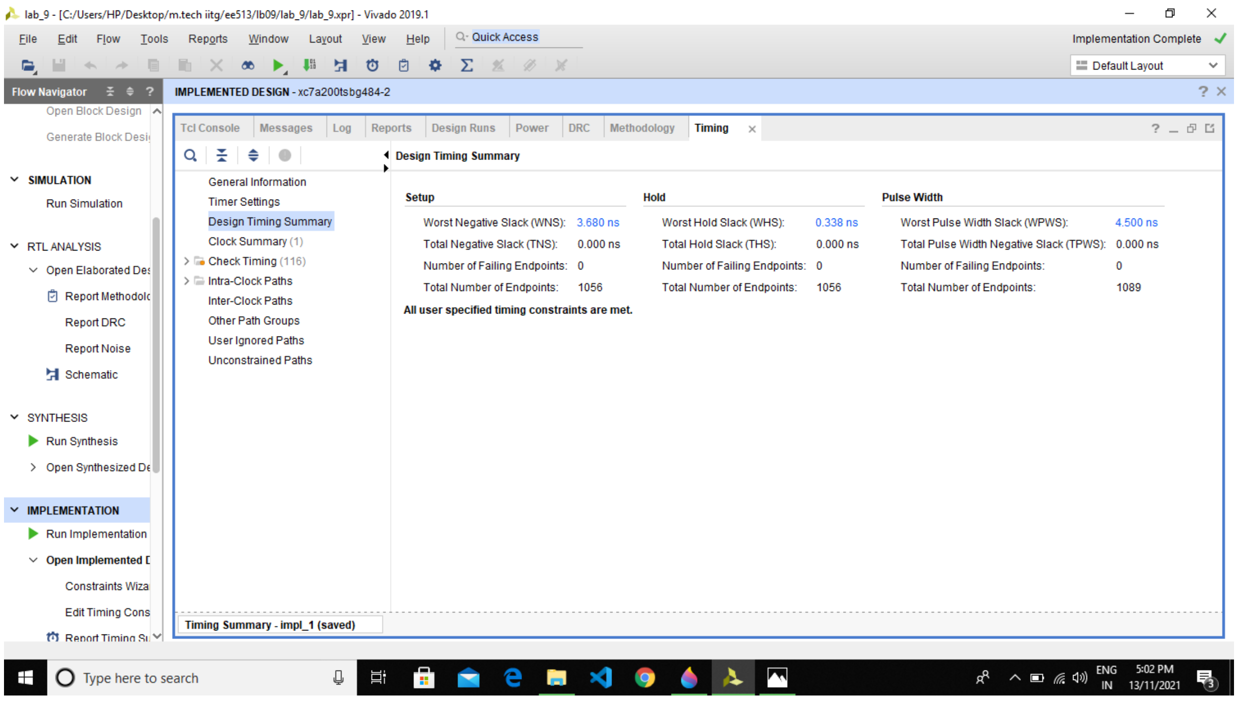
SLL,5 bit left shift:



waveform for all the function given in the table earlier:



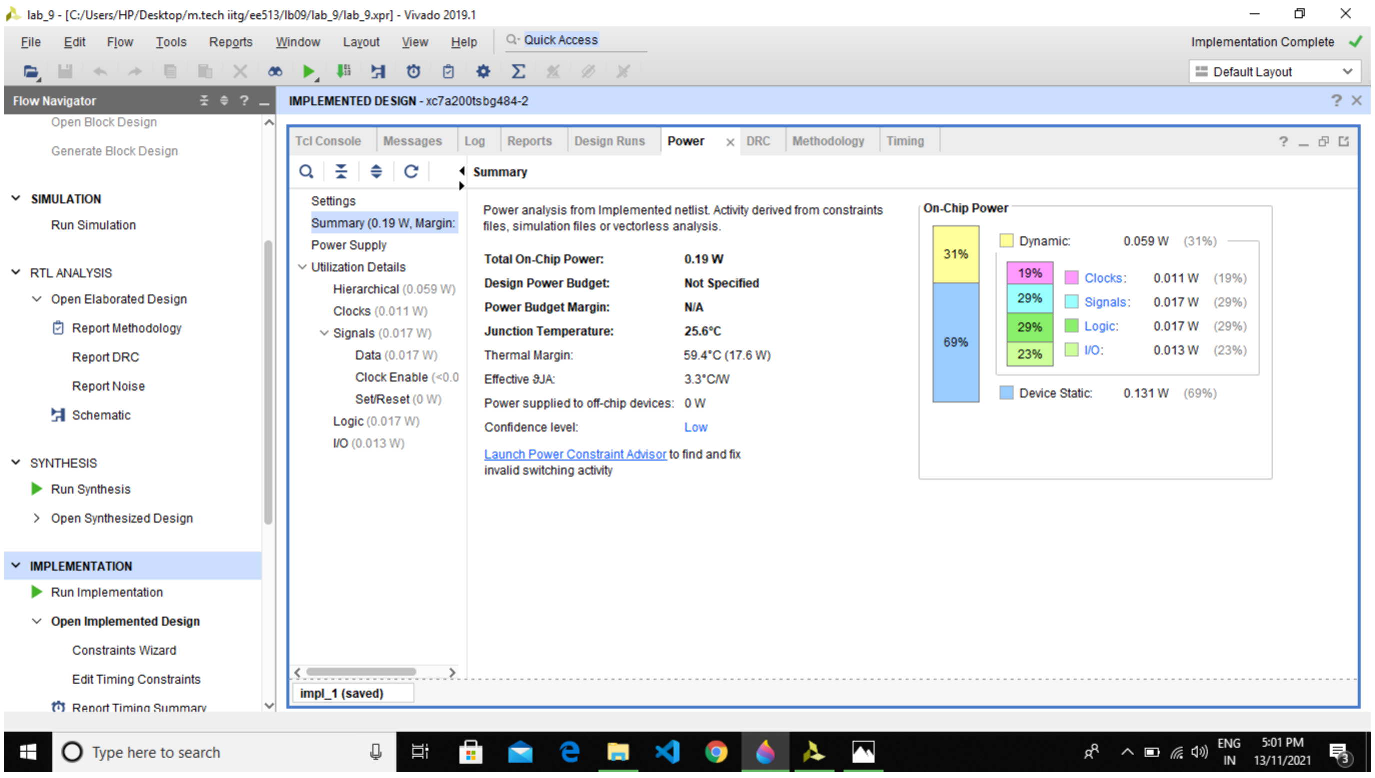
Delay Analysis:



Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.680 ns	Worst Hold Slack (WHS):	0.338 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	1056	Total Number of Endpoints:	1056	Total Number of Endpoints:	1089

All user specified timing constraints are met.

Power Analysis:



Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**0.19 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**25.6°C**
Thermal Margin:	59.4°C (17.6 W)
Effective ϑJA:	3.3°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

🟨	Dynamic:	0.059 W	(31%)
🟪	Clocks:	0.011 W	(19%)
🟦	Signals:	0.017 W	(29%)
🟩	Logic:	0.017 W	(29%)
🟩	I/O:	0.013 W	(23%)
🟦	Device Static:	0.131 W	(69%)

Result:

	No. of LUTs	No. of Flip Flops	Power	Delay
Processor	2617	1088	0.19 W	3.680 ns

Conclusion:

- I have successfully designed and implemented the 32 bit processor which will perform the given 32 bit instruction.
- I have taken a basic instruction in test bench for shifting left of 32 bit data. rs1 will be shift as much as the lower 5 bit of rs2.In this case lower bit of rs2 is 01101,It means rs1 32 bit data will shift 5 bit left and we can see in the waveform of SLL.
- I have also performed all the instruction given in the table and output wave form is uploaded in above page.
- No of LUTs and Flip Flops used are 2617 and 1088 respectively.
- Required power is 0.19 W and delay is 3.68 ns.