Group 28

Kandula Revanth (21CS10035)
Pola Gnana Shekar (21CS10052)

Code to find GCD:

We have used the example of GCD(18,12) = 6.

Instruction Number	Instruction
0	ADDi R1,zero,18
1	ADDI R2,zero,12
2	SUB R3,R1,R2
3	BZ R3, 10
4	BMI R3,7
5	SUB R1,R1,R2
6	BR 2
7	MOV R4,R2
8	MOV R2, R1
9	MOV R1,R4
10	MOV R7,R1

Code to do Booth Multiplication:

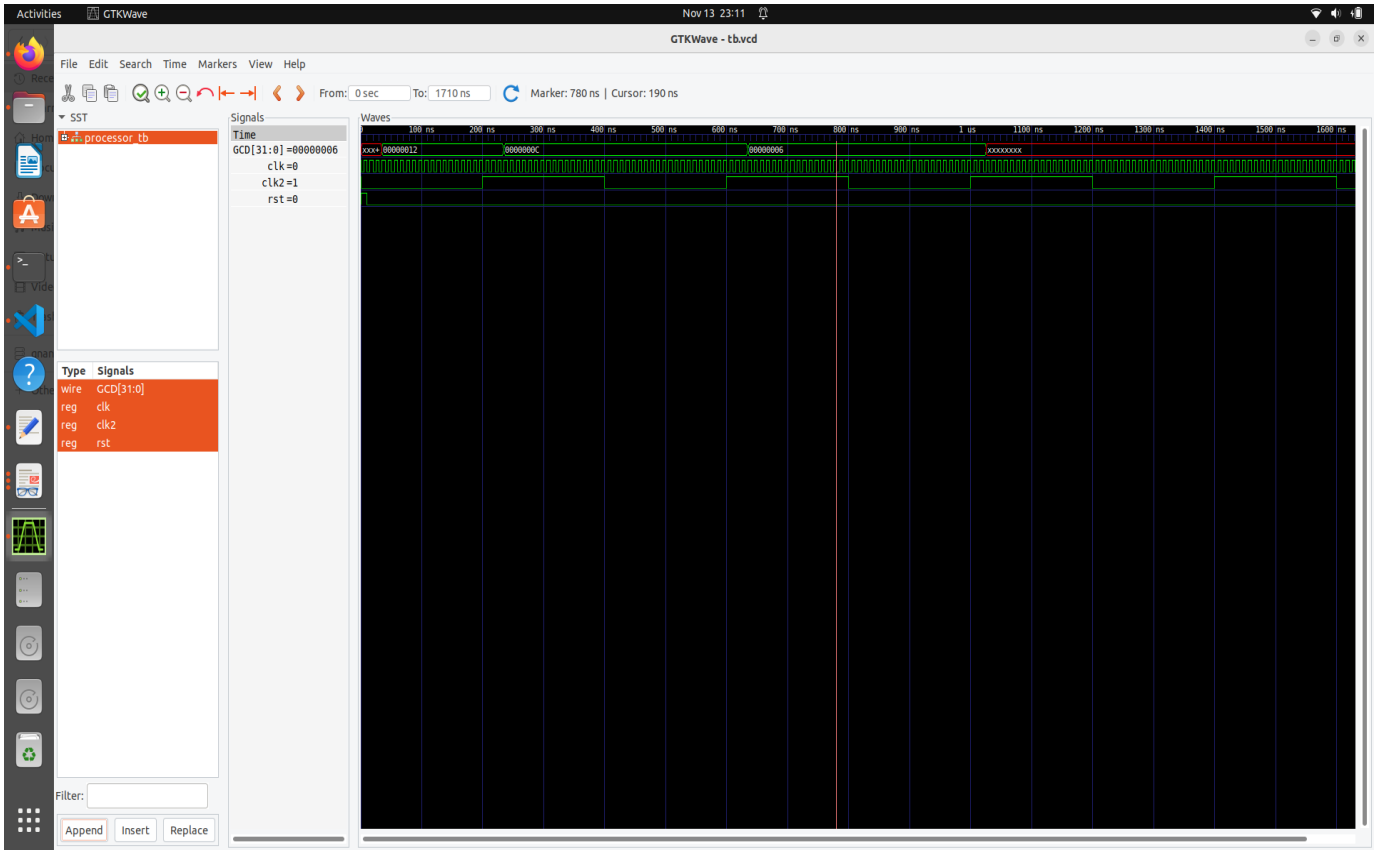
We have used the example of 2*8=16.

Instruction No	Instruction
19	ADDI R11, zero,1
20	ADDI R1,zero,2
21	ADDI R2,zero,8
22	ADDI R3,zero,0
23	ADDI R4,zero,4
24	ANDI R5,R2,1
25	BZ R5,27
26	SUB R3,R3,R1
27	SLL R1,R1,R11
28	SRA R2,R2,R11
29	BPL R4,24
30	MOV R8,R3

Note about files and code submitted:

- The test bench for GCD is in GCD_tb.v file and the corresponding instructions are in insMem.v
- Run this file to get the output.

Simulation for GCD:



Output for GCD:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

(base) gnanashekar2004@Shekar-Vostro-3590:~/Downloads/Verilog_Asgn_7_Part_5_Grp_28$ iverilog -o GCD.vvp GCD_tb.v
(base) gnanashekar2004@Shekar-Vostro-3590:~/Downloads/Verilog_Asgn_7_Part_5_Grp_28$ vvp GCD.vvp
VCD info: dumpfile GCD.vcd opened for output.
GCD =           6
(base) gnanashekar2004@Shekar-Vostro-3590:~/Downloads/Verilog_Asgn_7_Part_5_Grp_28$
```

Control signals used:

- readIn: controls reading of next instruction
  1 - read next instruction
  0 - don’t read
- MuxAlu1: select between regA and PC instruction
  1 - select regA
  0 - select PC
- MuxAlu2: select between regB and immediate value
  1 - select regB
  0 - select imm
- writeDM: write to Data Memory
  1 - write to Data Memory
  0 - don’t write

- ❖ readDM: read from Data Memory
  1 - read from Data Memory
  0 - don’t read
- ❖ muxWB: select between AluOut and value from LMD
  1 - select AluOut
  0 - select value from LMD
- ❖ writeReg: controls write action in register bank
  1 - write to register
  0 - don’t write
- ❖ rst: resets all the values
  1 - reset all
  0 - no need to reset
- ❖ branch: used to select type of branching (BR / BMI / BPL / BZ)
  000 - No branching
  001 - BR
  010 - BMI
  011 - BPL
  100 - BZ
- ❖ aluFunc: used to select the arithmetic operation that to be performed.
  0000 - ADD
  0001 - SUB
  0010 - NOT
  0011 - AND
  0100 - OR
  0101 - XOR
  0110 - SLL
  0111 - SRL
  1000 - SRA

Control Signal Table:

	readIn	MuxAlu1	MuxAlu2	writeDM	readDM	muxWB	writeReg	rst	branch	aluFunc

ADD	1	1	1	0	0	1	1	0	000	0000
SUB	1	1	1	0	0	1	1	0	000	0001
NOT	1	1	1	0	0	1	1	0	000	0010
AND	1	1	1	0	0	1	1	0	000	0011
OR	1	1	1	0	0	1	1	0	000	0100
XOR	1	1	1	0	0	1	1	0	000	0101
SLL	1	1	1	0	0	1	1	0	000	0110
SRL	1	1	1	0	0	1	1	0	000	0111
SRA	1	1	1	0	0	1	1	0	000	1000
ADDI	1	1	0	0	0	1	1	0	000	0000
SUBI	1	1	0	0	0	1	1	0	000	0001
NOTI	1	1	0	0	0	1	1	0	000	0010
ANDI	1	1	0	0	0	1	1	0	000	0011
ORI	1	1	0	0	0	1	1	0	000	0100
XORI	1	1	0	0	0	1	1	0	000	0101
SLLI	1	1	0	0	0	1	1	0	000	0110
SRLI	1	1	0	0	0	1	1	0	000	0111
LD	1	1	0	0	1	0	1	0	000	0000
ST	1	1	0	1	0	0	0	0	000	0000
LDSP	1	1	0	0	1	0	1	0	000	0000
STSP	1	1	0	1	0	0	0	0	000	0000
BR	1	0	0	0	0	0	0	0	001	0000
BMi	1	0	0	0	0	0	0	0	010	0000
BPL	1	0	0	0	0	0	0	0	011	0000
BZ	1	0	0	0	0	0	0	0	100	0000
PUSH	1	1	0	1	0	0	1	0	000	0001
POP	1	1	0	0	1	0	1	0	000	0000

CALL	1	1	0	1	0	0	1	0	000	0001
RET	1	1	0	0	1	0	1	0	000	0000
MOVE	1	1	0	0	0	1	1	0	000	0000
NOP	1	0	0	0	0	0	0	0	000	0000
HALT	0	0	0	0	0	0	0	1	000	0000

RTL instruction Table:

Instruction	RTL micro operations	Control Signals
ADD	- Source and Destination registers obtained from 32 bit Instruction. - Select the reg1 and reg2 as operands - Perform addition using aluFunc signal. - Select AluOut to write back. - Write back in register bank	- - MuxAlu1,MuxAlu2 - aluFunc - muxWB - writeReg
ADDI	- Source and Destination registers and immediate value obtained from 32 bit Instruction. - Select the reg1 and imm as operands - Perform addition using aluFunc signal. - Select AluOut to write back. - Write back in register bank	- - MuxAlu1,MuxAlu2 - aluFunc - muxWB - writeReg
LD	- Obtain source and Destination register from 32-bit instruction. - Read from data memory - Load from LMD for write back - Perform write back in register bank.	- - readDM - muxWB - writeReg
ST	- Obtain source and destination register from 32-bit instruction. - write to data memory	- writeDM
BMI	- Source register and offset are obtained from instruction. - Branch if less than zero is selected using branch - Update PC according to the condition.	- branch
MOVE	- Dest and src registers are obtained	- MuxAlu1,MuxAlu2

	from instruction. - Perform add operation with one operand as src register and zero register. - Select aluOut for write back. - Perform write back.	- aluFunc - muxWB - writeReg
CALL	- Immediate value from instruction. - Perform add operation to the PC and 4. - Store it in stack mem - Perform addition of PC and offset - Update this as next PC.	- writeDm
RET	- Perform read from stack mem. - Update this to PC.	- readDM

Data Path: