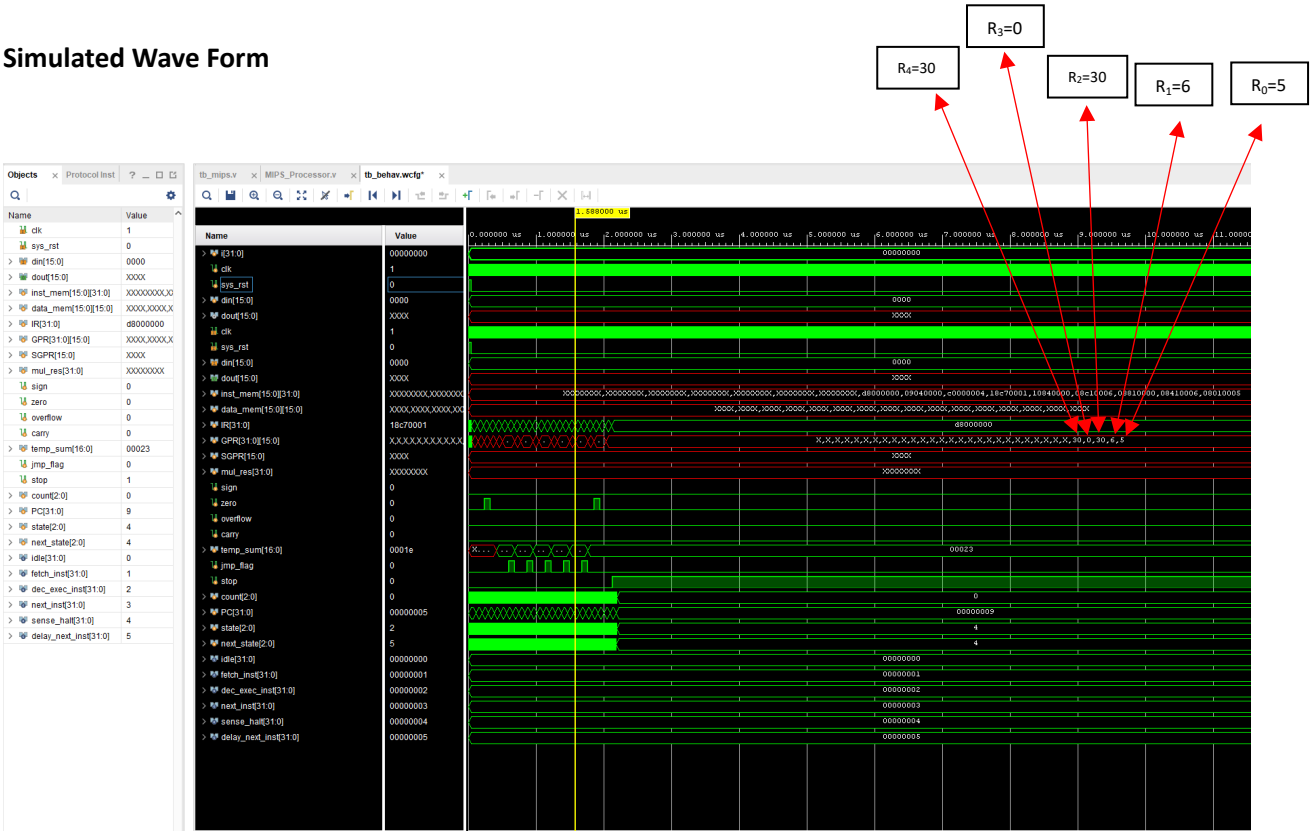
Refer to the Appendix section of this doc for instruction format and op codes of instructions.

MIPS Functional Verification

Perform Multiplication of 6 and 5 without MUL instruction

Inst_mem_address	Assembly Instruction	Binary inst code	Comments
0	Mov R_0, #5;	00001_00000_00000_1_0000_0000_0000_0101	Set Register, R_0=5
1	Mov R_1, #6;	00001_00001_00000_1_0000_0000_0000_0110	R_1=6
2	Mov R_2, #0;	00001_00010_00000_1_0000_0000_0000_0000	R_2=0
3	Mov R_3, #6;	00001_00011_00000_1_0000_0000_0000_0110	R_3=6
4	ADD R_2,R_2,R_0	00010_00010_00010_0_0000_0000_0000_0000	Final value of R_2=30
5	SUB R_3,R_3, #1;	00011_00011_00011_1_0000_0000_0000_0001	R_3=5,4,3,…,0
6	JNZ @4	11000_00000_00000_0_0000_0000_0000_0100	Jump to inst_mem_addr 4 when 'jumpnozero is high
7	Mov R_4,R_2	00001_00100_00010_0_0000_0000_0000_0000	Final value of R_4=30
8	HALT	11011_00000_00000_0_0000_0000_0000_0000	

Simulated Wave Form



Refer to the Appendix section of this doc for instruction format and op codes of instructions.

Appendix

Instruction Format for the Instruction Register

IR <--ir[31:27]--><--ir[26:22]--><--ir[21:17]--><--ir[16]--><--ir[15:11]--><--ir[10:0]-->

Fields <--- oper --><-- rdest --><-- rsrc1 --><--modesel--><-- rsrc2 --><--unused -->

Fields <--- oper --><-- rdest --><-- rsrc1 --><--modesel--><-- immediate_date -->

OP Codes for different operation types

Arithmetic	Op code
movsgpr	5'b00000
mov	5'b00001
add	5'b00010
sub	5'b00011
mul	5'b00100

Logical	Op code
or	5'b00101
and	5'b00110
xor	5'b00111
xnor	5'b01000
nand	5'b01001
nor	5'b01010
not	5'b01011

Load & Store	Op code
storereg	5'b01101
storedin	5'b01110
senddout	5'b01111
sendreg	5'b10001

Refer to the Appendix section of this doc for instruction format and op codes of instructions.

Jump and Branch	Op code
jump	5'b10010
jcarry	5'b10011
jnocarry	5'b10100
jsign	5'b10101
jnosign	5'b10110
jzero	5'b10111
jnozero	5'b11000
joverflow	5'b11001
jnooverflow	5'b11010

Halt	Op code
halt	5'b11011