Report

Group 8
Junda Tu, Lingtao Shui, Jianbo Liu

Control unit

The input of the control unit is the 32-bit instruction. And there are 10 outputs of the control unit, the ports are RegDst, ALUSrc, MemtoReg, RegW, MemR, MemW, Branch, Bne, Bgtz and the 4-bit ALUCtr.

When RegDst outputs 0, the register destination number for the write register comes from the rt field. Otherwise it comes from the rd field.
When ALUSrc outputs 0, the second ALU operand comes from the second register file output. Otherwise, the second operand is the sign-extended.
When MemtoReg outputs 0, the value fed to the register Write data input comes from the ALU. Otherwise, the value comes from the memeory.
When RegW outputs 1, the register is written with the value on the Write data input.
When MemR outputs 1. Data memory contents designated by the address input are put on the Read data output.
When MemW outputs 1, Data memory contents designated by the address input are replaced by the value on the Write data input.
When Branch outputs 1, IF unit will check Zero flag to determine branch or not.
When Bne outputs 1, IF unit will check Zero flag to determine branch or not.
When Bgtz outputs 1, IF unit will check Zero flag and R[s]' MSB to determine branch or not.

And the control can be divided into 2 parts. One is the main control and another is the alu control.

Main control

The input of the main control 26-31 bit instruction. And the outputs are RegDst, ALUSrc, MemtoReg, RegW, MemR, MemW, Branch, Bne, Bgtz and 2-bit ALU_op which is used to control the alu control.

And table below is the truth table of the main control.

main control truth table												
	OP	RegDst	ALUSrc	MemtoReg	RegW	MemR	MemW	Branch	BNE	BGTZ	ALU_op(1)	ALU_op(0)
R-type	000000	1	0	0	1	0	0	0	0	0	1	0

LW	100011	0	1	1	1	1	0	0	0	0	0	0
SW	101011	x	1	x	0	0	1	0	0	0	0	0
ADDI	001000	0	1	0	1	0	0	0	0	0	0	0
BEQ	000100	x	0	x	0	0	0	1	0	0	0	1
BNE	000101	x	0	x	0	0	0	0	1	0	0	1
BGTZ	000111	x	0	x	0	0	0	0	0	1	0	1

As we can see there are 7 different types of operations.
When the operation is R-type, the register destination number for the write register comes from the rd field. the second ALU operand comes from the second register file output. the value fed to the register Write data input comes from the ALU. And the register is written with the value on the Write data input. And R-type will not use the memory.

When the operation is LW, the register destination number for the write register comes from the rt field. And the second operand comes from the sign-extended. It will read the data from the memory and write into the register.

When the operation is SW, it do not need the register. And the second operand comes from the sign-extended. And write the output of the alu into memory.

When the operation is ADDI, the register destination number for the write register comes from the rt field. The second operand coms from the sign-extended. And need to write the output data of the alu into register.

When the operation is BEQ, only the IF unit and zero flag matter. If zero flag is set then pc = pc + 4 + offset * 4.
When the operation is BNE, if zero flag is not set then pc = pc + 4 + offset * 4.
When the operation is BGTZ, if zero flag is not set and [Rs] is greater than zero then pc = pc + 4 + offset * 4.

ALU control

The inputs of the ALU control are instruction of 0 to 5 bit, and the ALU_op. And the output of the ALU control is the 4 bits ALUCtr.
And table below is the truth table of the ALU control.

ALU _op	Func	ALUCtr	operation

00	xxxxxx	0011	add
00	xxxxxx	0011	add
01	xxxxxx	0111	sub
01	xxxxxx	0111	sub
01	xxxxxx	0111	sub
00	xxxxxx	0011	add
10	100000	0011	add
10	100001	0011	add
10	100010	0111	sub
10	100011	0111	sub
10	100100	0000	and
10	100101	0001	or
10	000000	1000	sll
10	101010	0100	slt
10	101011	0101	sltu

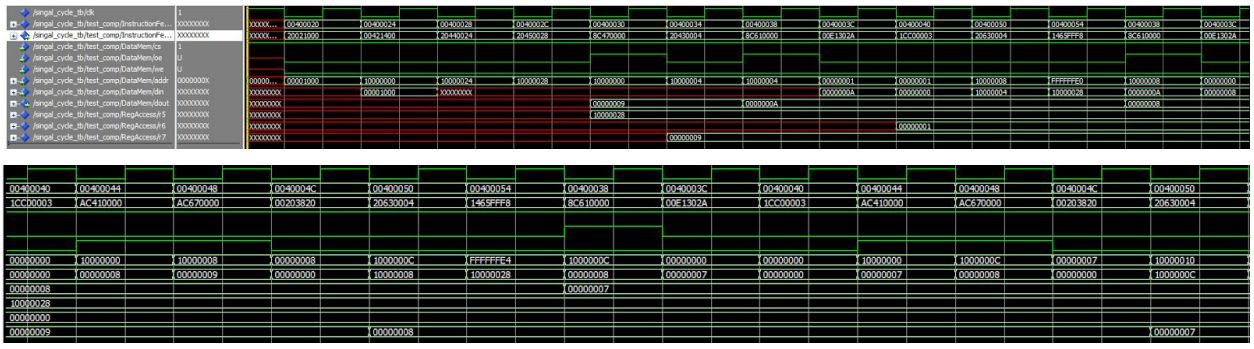
As we can see, the ALU_op data of the R-type operation is 10. And ALU_op of the SW,LW,ADDI are 00. The ALU_op of the BEQ,BNE,BGTZ are 01. The ALUCtr is just the signal that control the ALU. Then we just combine this two parts and get the whole control unit.

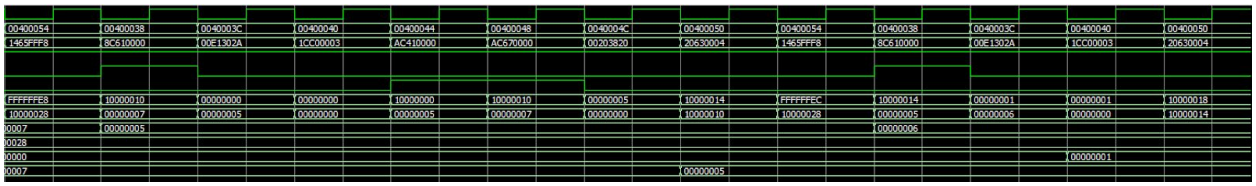
Problem in the Project

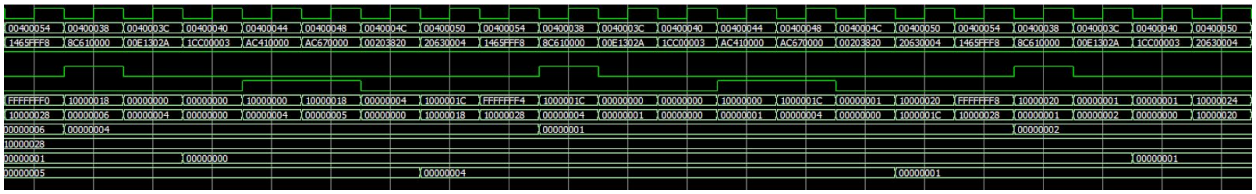
One problem is how to use the clock signal in Reg_file. Since we only write back at falling edge of the clock, we use a and gate to sync the write back and the clk.
Another problem is we need to add two more signals to the control unit to distinguish between beq, bne and bgtz. They are all branch but have different branch conditions.

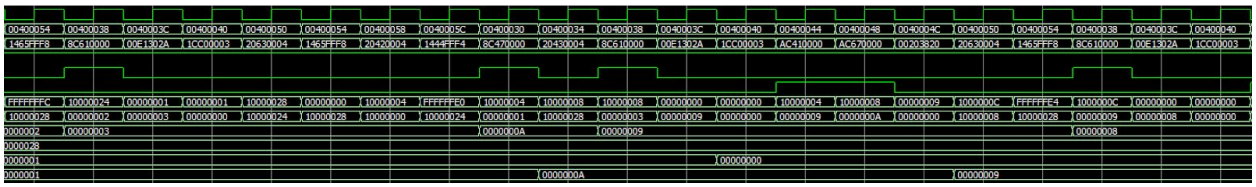
Trace

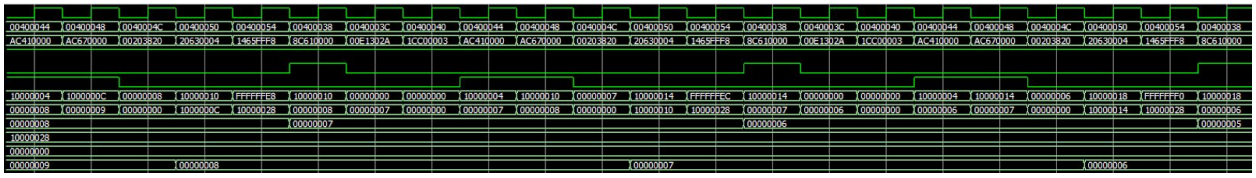
1. Sort_corrected_branch

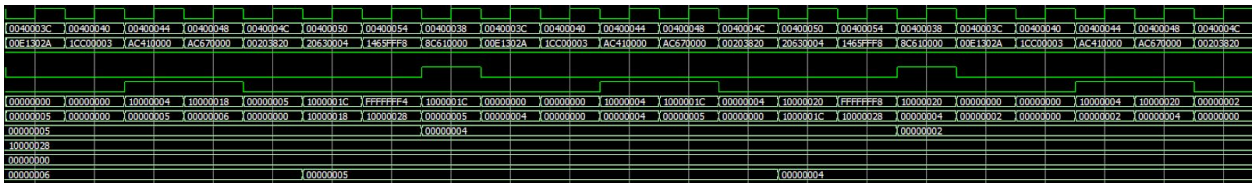


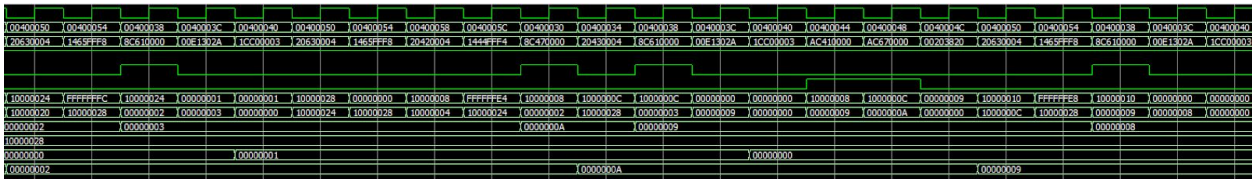

00400054 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400050
1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 20630004
FFFFFFE8 10000010 00000000 00000000 10000000 10000010 00000005 10000014 FFFFFFEC 10000014 00000001 00000001 10000018
10000028 00000007 00000005 00000000 00000005 00000007 00000000 10000010 10000028 00000005 00000006 00000000 10000014
0007 00000005 00000006
0028
0000 00000001
0007 00000005

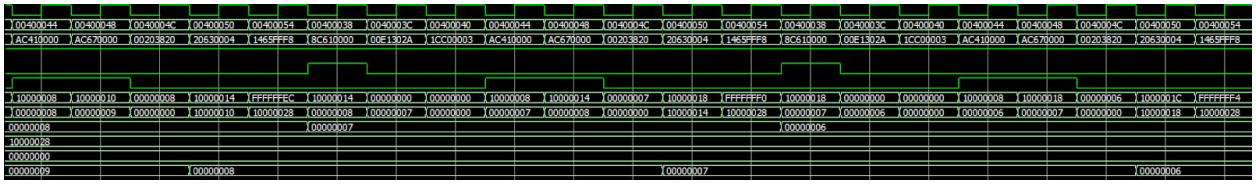

00400054 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400050
1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 20630004
FFFFFFF0 10000018 00000000 00000000 10000000 10000018 00000004 1000001C FFFFFFF4 1000001C 00000000 00000000 10000000 1000001C 00000001 10000020 FFFFFFF8 10000020 00000001 00000001 10000024
10000028 00000006 00000004 00000000 00000004 00000005 00000000 10000018 10000028 00000004 00000001 00000000 00000001 00000004 00000000 1000001C 10000028 00000001 00000002 00000000 10000020
00000006 00000004 00000001 00000002
10000028
00000001 00000000 00000001
00000005 00000004 00000001

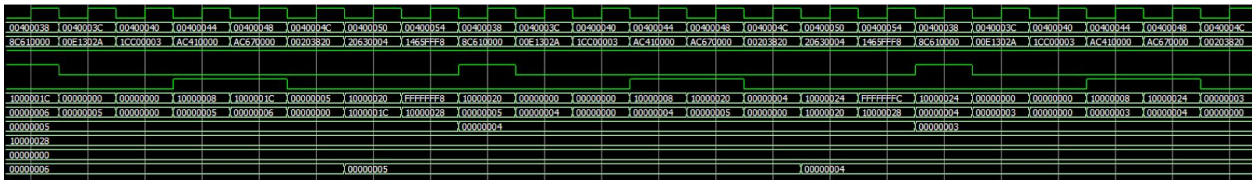

00400054 00400038 0040003C 00400040 00400050 00400054 00400058 0040005C 00400030 00400034 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040
1465FFF8 8C610000 00E1302A 1CC00003 20630004 1465FFF8 20420004 1444FFF4 8C470000 20430004 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003
FFFFFFFC 10000024 00000001 00000001 10000028 00000000 10000004 FFFFFFE0 10000004 10000008 10000008 00000000 00000000 10000004 10000008 00000009 1000000C FFFFFFE4 1000000C 00000000 00000000
10000028 00000002 00000003 00000000 10000024 10000028 10000000 10000024 00000001 10000028 00000003 00000009 00000000 00000009 0000000A 00000000 10000008 10000028 00000009 00000008 00000000
00000002 00000003 0000000A 00000009 00000008
00000038
00000001 00000000
00000001 0000000A 00000009

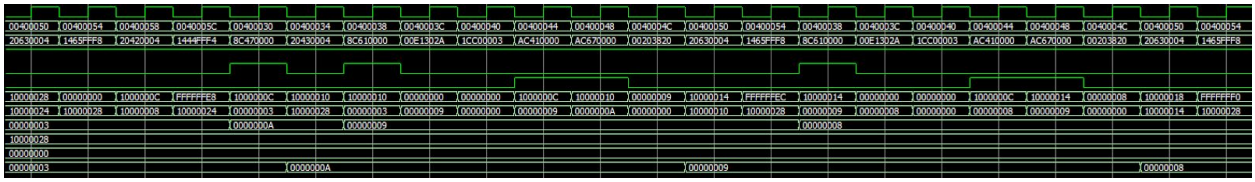

00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038
AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000
10000004 1000000C 00000008 10000010 FFFFFFE8 10000010 00000000 00000000 10000004 10000010 00000007 10000014 FFFFFFEC 10000014 00000000 00000000 10000004 10000014 00000006 10000018 FFFFFFF0 10000018
00000008 00000009 00000000 1000000C 10000028 00000008 00000007 00000000 00000007 00000008 00000000 10000010 10000028 00000007 00000005 00000000 00000005 00000007 00000000 10000014 10000028 00000006
00000008 00000007 00000006 00000005
10000028
00000000
00000009 00000008 00000007 00000006

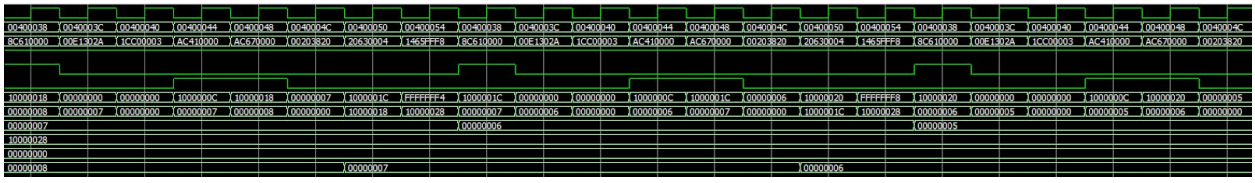

0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400044 00400048 0040004C
00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820
00000000 00000000 10000004 10000018 00000005 1000001C FFFFFFF4 1000001C 00000000 00000000 10000004 1000001C 00000004 10000020 FFFFFFF8 10000020 00000000 00000000 10000004 10000020 00000002
00000005 00000000 00000005 00000006 00000000 10000018 10000028 00000005 00000004 00000000 00000004 00000005 00000000 1000001C 10000028 00000004 00000002 00000000 00000002 00000004 00000000
00000005 00000004 00000002
10000028
00000000
00000006 00000005 00000004

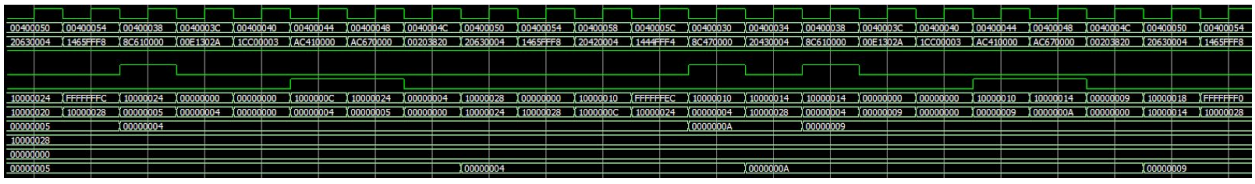

00400050 00400054 00400038 0040003C 00400040 00400050 00400054 00400058 0040005C 00400030 00400034 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040
20630004 1465FFF8 8C610000 00E1302A 1CC00003 20630004 1465FFF8 20420004 1444FFF4 8C470000 20430004 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003
10000024 FFFFFFFC 10000024 00000001 00000001 10000028 00000000 10000008 FFFFFFE4 10000008 1000000C 1000000C 00000000 00000000 10000008 1000000C 00000009 10000010 FFFFFFE8 10000010 00000000 00000000
10000020 10000028 00000002 00000003 00000000 10000024 10000028 10000004 10000024 00000002 10000028 00000003 00000009 00000000 00000009 0000000A 00000000 1000000C 10000028 00000009 00000008 00000000
00000002 00000003 0000000A 00000009 00000008
10000028
00000000 00000001 00000000
00000002 0000000A 00000009

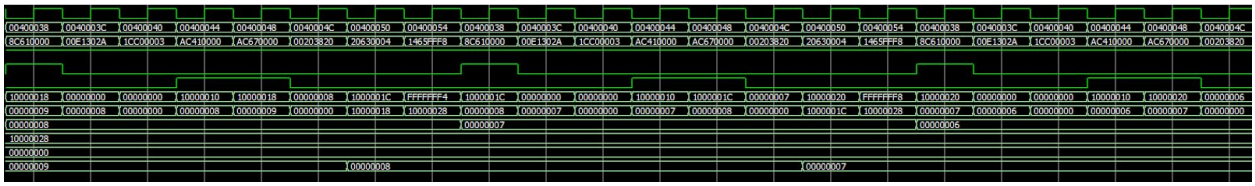

00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054
AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8
10000008 10000010 00000008 10000014 FFFFFFEC 10000014 00000000 00000000 10000008 10000014 00000007 10000018 FFFFFFF0 10000018 00000000 00000000 10000008 10000018 00000006 1000001C FFFFFFF4
00000008 00000009 00000000 10000010 10000028 00000008 00000007 00000000 00000007 00000008 00000000 10000014 10000028 00000007 00000005 00000000 00000005 00000007 00000000 10000018 10000028
00000008 00000007 00000006
10000028
00000000
00000009 00000008 00000007 00000006

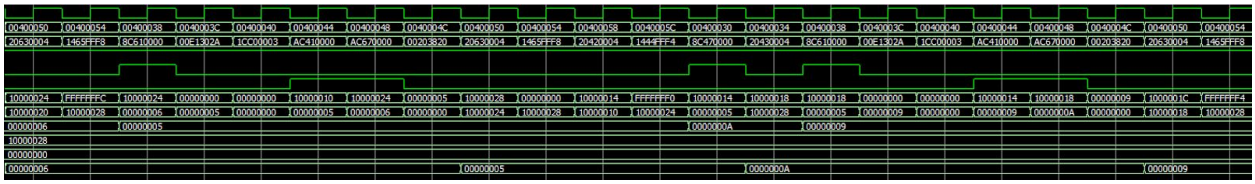

00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400044 00400048 0040004C 00400050 00400054 00400038 0040003C 00400040 00400044 00400048 0040004C
8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820 20630004 1465FFF8 8C610000 00E1302A 1CC00003 AC410000 AC670000 00203820
1000001C 00000000 00000000 10000008 1000001C 00000005 10000020 FFFFFFF8 10000020 00000000 00000000 10000008 10000020 00000004 10000024 FFFFFFFC 10000024 00000000 00000000 10000008 10000024 00000003
00000006 00000005 00000000 00000005 00000006 00000000 1000001C 10000028 00000005 00000004 00000000 00000004 00000005 00000000 10000020 10000028 00000004 00000003 00000000 00000003 00000004 00000000
00000005 00000004 00000003
10000028
00000000
00000006 00000005 00000004

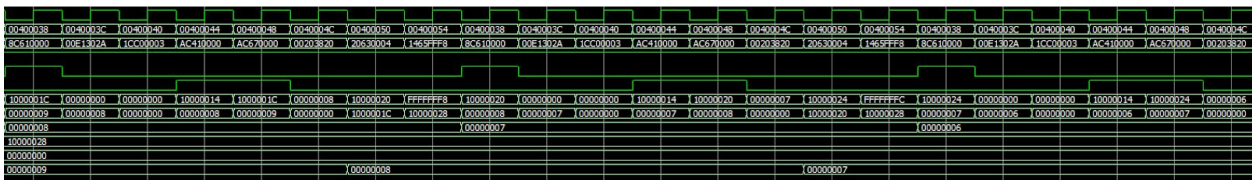


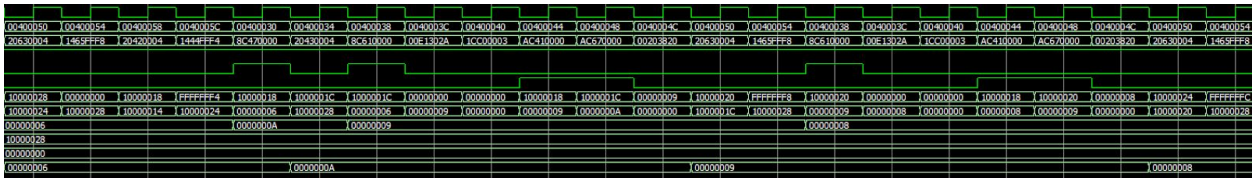


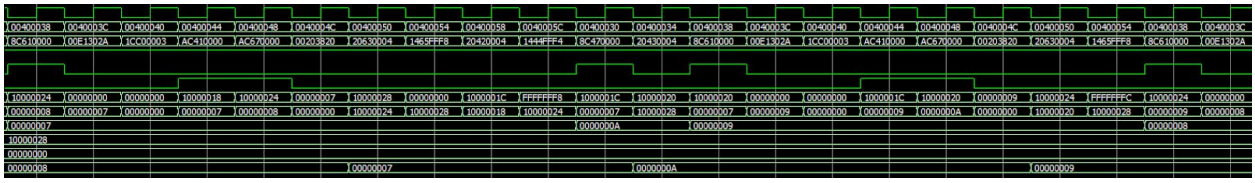


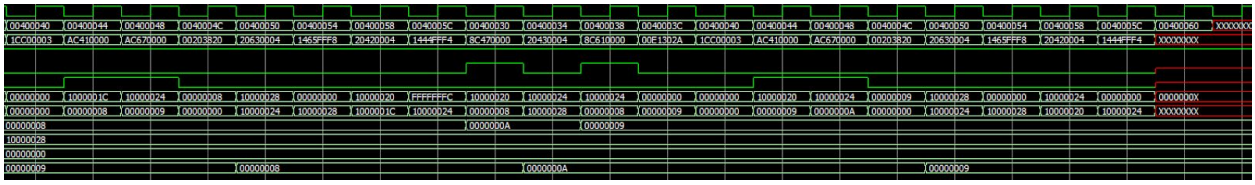




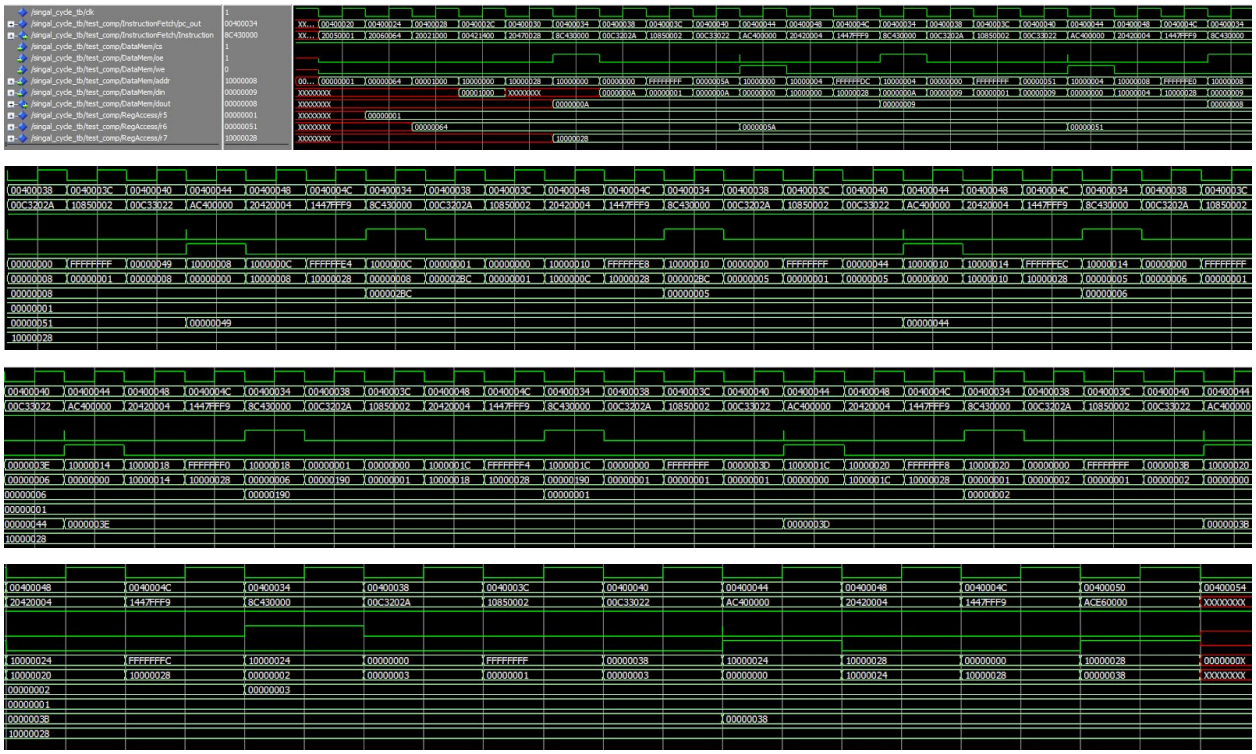








2. Bills_branch



3.unsigned_sum