

EDA Edit code - EDA Playground

EDA EPWave Waveform Viewer

← → ↺

edaplayground.com

☆ 📁 📄 🖱

Gmail YouTube Maps

EDA playground

New Run Save*

Resources ▾ Community ▾ Help ▾ Playgrounds ▾ Profile ▾

Brought to you by

DOULOS

DOulos does not endorse training material from other suppliers on EDA Playground.

Languages & Libraries

Testbench + Design

SystemVerilog/Verilog ▾

UVM / OVM ?

None ▾

Other Libraries ?

None ▾

OVl

SVUnit

☐ Enable TL-Verilog ⓘ

☐ Enable Easier UVM ⓘ

☐ Enable VUnit ⓘ

Tools & Simulators ?

Icarus Verilog 12.0 ▾

Compile Options

-Wall -g2012

Run Options

Run Options

☐ Use run.bash shell script

☒ Open EPWave after run

☐ Show output file after run

☐ Download files after run

Examples

testbench.sv

1 //Program 1 to be runned on the designed Processor, (testbench)

2 //Bipriti Deb Sarma

3 module test_mips32;

4 reg clk1, clk2;

5 integer k;

6

7 // Instantiate the MIPS32 processor module

8 pipe_MIPS32 mips (clk1, clk2);

9

10 initial begin

11 clk1 = 0;

12 clk2 = 0;

13

14 // Generating two-phase clock

15 repeat (20) begin

16 #5 clk1 = 1;

17 #5 clk1 = 0;

18 #5 clk2 = 1;

19 #5 clk2 = 0;

20 end

21 end

22

23 initial begin

24 // Initialize register values

25 for (k = 0: k < 32: k = k + 1)

design.sv

1 // RISC PROCESSOR - MIPS32 DESIGN CODE

2 //Bipriti Deb Sarma

3

4 module pipe_MIPS32 (clk1, clk2); // Two-phase clock

5 input clk1, clk2;

6

7 reg [31:0] PC, IF_ID_IR, IF_ID_NPC;

8 reg [31:0] ID_EX_IR, ID_EX_NPC, ID_EX_A, ID_EX_B, ID_EX_Imm;

9 reg [2:0] ID_EX_type, EX_MEM_type, MEM_WB_type;

10 reg [31:0] EX_MEM_IR, EX_MEM_ALUOut, EX_MEM_B;

11 reg EX_MEM_cond;

12 reg [31:0] MEM_WB_IR, MEM_WB_ALUOut, MEM_WB_LMD;

13 reg [31:0] regbank [0:31];

14 reg [31:0] mem [0:1023];

15 // Instruction opcodes and parameter declarations

16 parameter ADD = 6'b000000, SUB = 6'b000001, AND = 6'b000010, OR = 6'b000011,

17 SLT = 6'b000100, MUL = 6'b000101, HLT = 6'b111111, LW = 6'b001000,

18 SW = 6'b001001, ADDI = 6'b001010, SUBI = 6'b001011, SLTI = 6'b001100,

19 BNEQZ = 6'b001101, BEQZ = 6'b001110;

20

21 parameter RR_ALU = 3'b000, RM_ALU = 3'b001, LOAD = 3'b010, STORE = 3'b011,

22 BRANCH = 3'b100, HALT = 3'b101;

23

24 reg HALTED; // Set after HLT instruction is completed (in WB stage)

25 reg TAKEN_BRANCH; // Required to disable instructions after branch

Log Share

R0 = 0

R1 = 10

R2 = 20

R3 = 25

R4 = 30

R5 = 55

Finding VCD file...

./mips.vcd

[2024-09-20 16:52:44 UTC] Opening EPWave...

Done

EDA Playground

EPWave Waveform Viewer

edaplayground.com/launchEppwave

EPWave beta

Load

Save

Examples

About

Apps

Rcc1bba34-2697-4b85-be74-d3e9e

EDA Playground

https://www.edaplayground.com/

From: 0s To: 400s

Get Signals

Radix

100%

499,999,999,913s

clk1

1

clk2

0

EX_MEM_ALUOut[31:0]

7

NaN

10

20

25

7

30

7

55

EX_MEM_B[31:0]

NaN

NaN

EX_MEM_IR[31:0]

128

NaN

671154186

671219732

671285273

216496128

2236416

216496128

8595456

4227858432

EX_MEM_type[2:0]

0

NaN

1

0

5

ID_EX_A[31:0]

7

NaN

0

7

10

7

30

0

*N

ID_EX_B[31:0]

7

NaN

1

2

3

7

20

7

25

0

*N

ID_EX_IR[31:0]

128

NaN

671154186

671219732

671285273

216496128

2236416

216496128

8595456

4227858432

*N

ID_EX_Imm[31:0]

720

NaN

10

20

25

30720

8192

30720

10240

0

*N

ID_EX_NPC[31:0]

4

NaN

1

2

3

4

5

6

7

8

9

10

IF_ID_IR[31:0]

128

NaN

671154186

671219732

671285273

216496128

2236416

216496128

8595456

4227858432

NaN

IF_ID_NPC[31:0]

5

NaN

1

2

3

4

5

6

7

8

9

10

MEM_WB_ALUOut[31:0]

25

NaN

10

20

25

7

30

7

55

MEM_WB_IR[31:0]

273

NaN

671154186

671219732

671285273

216496128

2236416

216496128

8595456

*2

MEM_WB_LMD[31:0]

NaN

NaN

PC[31:0]

5

0

1

2

3

4

5

6

7

8

9

10

Brought to you by

DOULOS

EDA Edit code - EDA Playground

EDA EPWave Waveform Viewer

← → ↺

edaplayground.com

🔍 ☆ 📁 📄 🎧

Gmail

YouTube

Maps

playground

NewRunSave*

ResourcesCommunityHelpPlaygroundsProfile

Brought to you by

DOULOS

DOULOS does not endorse training material from other suppliers on EDA Playground.

Languages & Libraries

Testbench + Design

SystemVerilog/Verilog

UVM / OVM

None

Other Libraries

NoneOVLVUnit

☐ Enable TL-Verilog

☐ Enable Easier UVM

☐ Enable VUnit

Tools & Simulators

Icarus Verilog 12.0

Compile Options

-Wall -g2012

Run Options

Run Options

☐ Use run.bash shell script

☒ Open EPWave after run

☐ Show output file after run

☐ Download files after run

Examples

using EDA Playground VHDL

Verilog/SystemVerilog

testbench.sv

```
1 //Program 2 to be runned on the designed Processor, (testbench)
2 //Bipriti Deb Sarma
3 module test_mips32;
4   reg clk1, clk2;
5   integer k;
6
7   // Instantiate the MIPS processor module
8   pipe_MIPS32 mips (clk1, clk2);
9
10  // Clock generation block: generate two-phase clock
11  initial
12  begin
13    clk1 = 0;
14    clk2 = 0;
15    repeat(50) // Generating two-phase clock for 50 cycles
16    begin
17      #5 clk1 = 1;
18      #5 clk1 = 0;
19      #5 clk2 = 1;
20      #5 clk2 = 0;
21    end
```

design.sv

```
1 // RISC PROCESSOR - MIPS32 DESIGN CODE
2 //Bipriti Deb Sarma
3
4 module pipe_MIPS32 (clk1, clk2); // Two-phase clock
5   input clk1, clk2;
6
7   reg [31:0] PC, IF_ID_IR, IF_ID_NPC;
8   reg [31:0] ID_EX_IR, ID_EX_NPC, ID_EX_A, ID_EX_B, ID_EX_Imm;
9   reg [2:0] ID_EX_type, EX_MEM_type, MEM_WB_type;
10  reg [31:0] EX_MEM_IR, EX_MEM_ALUout, EX_MEM_B;
11  reg EX_MEM_cond;
12  reg [31:0] MEM_WB_IR, MEM_WB_ALUout, MEM_WB_LMD;
13  reg [31:0] regbank [0:31];
14  reg [31:0] mem [0:1023];
15  // Instruction opcodes and parameter declarations
16  parameter ADD = 6'b000000, SUB = 6'b000001, AND = 6'b000010, OR = 6'b000011,
17            SLT = 6'b000100, MUL = 6'b000101, HLT = 6'b111111, LW = 6'b001000,
18            SW = 6'b001001, ADDI = 6'b001010, SUBI = 6'b001011, SLTI = 6'b001100,
19            BNEQZ = 6'b001101, BEQZ = 6'b001110;
20
21  parameter RR_ALU = 3'b000, RM_ALU = 3'b001, LOAD = 3'b010, STORE = 3'b011,
```

LogShare

[2024-09-20 16:55:59 UTC] iverilog '-wall' '-g2012' design.sv testbench.sv && unbuffer vvp a.out

VCD info: dumpfile mips.vcd opened for output.

Time: 0 | R2: 2

Time: 67 | R2: 1

Time: 147 | R2: 7

Time: 247 | R2: 42

Time: 347 | R2: 210

Time: 447 | R2: 840

Time: 547 | R2: 2520

Time: 647 | R2: 5040

Time: 747 | R2: 5040

Mem[200] = 7, Mem[198] = 5040

testbench.sv:68: \$finish called at 3000 (1s)

Finding VCD file...

./mips.vcd

[2024-09-20 16:56:00 UTC] Opening EPWave...

EDA Edit code - EDA Playground

EDA EPWave Waveform Viewer

edaplayground.com/launchEpwave

Gmail YouTube Maps

EPWave beta Load Save Examples About

Apps Rcc1bba34-2697-4b85-be74-d3e9e

EDA Playground

https://www.edaplayground.com/

From: 0s To: 3,000s

Get Signals Radix 100% 499,999,999,663s

		0	100	200	300	400	500	600	700																																	
	c1k1	0																																								
	c1k2	1																																								
+	EX_MEM_ALUout[31:0]	210	NaN	200	1	20	200	20	7	6	20	5	198	42	5	20	5	198	210	4	20	5	198	840	3	20	5	198	2520	2	20	5	198	5040	1	20	5	198	5040			
+	EX_MEM_B[31:0]	42	NaN																																							
+	EX_MEM_IR[31:0]	328	NaN	*200	*713	*312	*040	*312	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328			
+	EX_MEM_type[2:0]	0	NaN	1	0	2	0	1	0	4	3	0	1	0	4	3	0	1	0	4	3	0	1	0	4	3	0	1	0	4	3	0	1	0	4	3	0	1	0	4	3	0
+	ID_EX_A[31:0]	5	NaN	0	20	200	20	1	7	20	6	200	7	6	20	5	200	42	5	20	4	200	210	4	20	3	200	840	3	20	2	200	2520	2	20	1	200	5040	1			
+	ID_EX_B[31:0]	5	NaN	10	2	20	3	20	7	20	0	7	6	20	0	42	5	20	0	210	4	20	0	840	3	20	0	2520	2	20	0	5040	1									
+	ID_EX_IR[31:0]	569	NaN	*200	*713	*312	*040	*312	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569		
+	ID_EX_Imm[31:0]	1	NaN	200	1	*720	0	*720	4096	1	*720	*292	*294	4096	1	*720	*292	*294	4096	1	*720	*292	*294	4096	1	*720	*292	*294	4096	1	*720	*292	*294	4096	1	*720	*292	*294	4096	1		
+	ID_EX_NPC[31:0]	7	NaN	1	2	3	4	5	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7		
+	IF_ID_IR[31:0]	569	*	*200	*713	*312	*040	*312	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569		
+	IF_ID_NPC[31:0]	7	*	1	2	3	4	5	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7		
+	MEM_WB_ALUout[31:0]	210	NaN	200	1	20																																				
+	MEM_WB_IR[31:0]	328	NaN	*200	*713	*312	*040	*312	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569	*312	*220	*902	*328	*569		
+	MEM_WB_LMD[31:0]	7	NaN																																							
+	PC[31:0]	7	0	1	2	3	4	5	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7	8	9	10	6	7		

Brought to you by DOULOS