

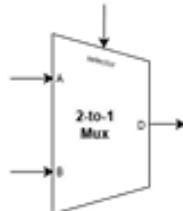


Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Arsitektur Sistem Komputer (EL3011)
Tahun / Semester : 2025-2026 / Ganjil
Modul : SYNTHESIZABLE RISC-V (RV32I) MICROPROCESSOR
BAGIAN II : CONTROL UNIT (CU), IMMEDIATE
SELECTOR, BRANCHER, PROGRAM COUNTER (PC),
4-ADDER, DAN 2-TO-1 MUX GENERIK
Nama Asisten / NIM : _____
Nama Praktikan / NIM : William Anthony / 13223048

Tugas Pendahuluan

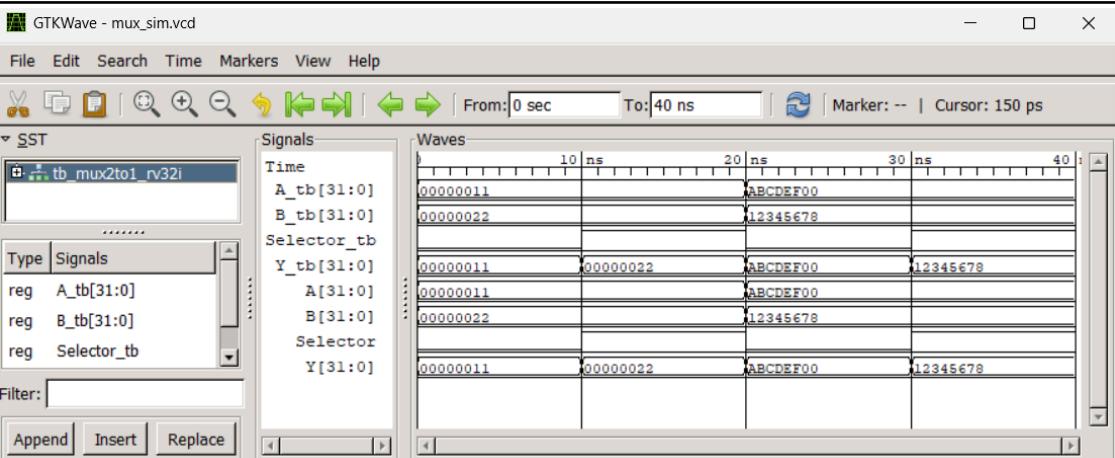
1. Rancang dan verifikasikan komponen **multiplexer 2-to-1** generik selebar 32-bit yang memilih keluaran Y dari dua masukan A dan B berdasarkan sinyal selektor sel ($0 \rightarrow A, 1 \rightarrow B$). Desain harus **synthesizable** dan diuji dengan **simulasi fungsional** serta **simulasi timing** (*gate-level* + anotasi penundaan). Cantumkan kode dan tangkapan layar *waveform* dari kedua simulasi.



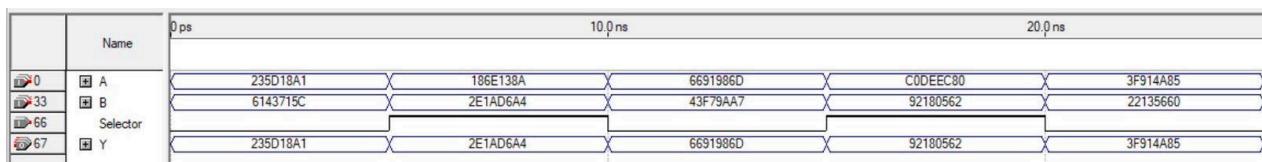
Kode

```
// Praktikum EL3011 Arsitektur Sistem Komputer
// Modul      : 2
// Percobaan  : 1
// Tanggal    : 19 November 2025
// Nama (NIM) 1 : William Anthony (13223048)
// Nama (NIM) 2 : Agita Trinanda Ilmi (13223003)
// Nama File   : mux2to1_rv32i.v
// Deskripsi   : Desain multiplexer 2-ke-1 generik 32-bit

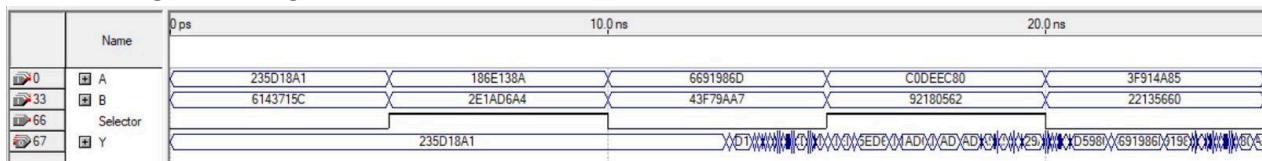
module mux2to1_rv32i (
    input wire [31:0] A,
    input wire [31:0] B,
    input wire Selector,
    output reg [31:0] Y
);
    always @(*) begin
        if (Selector == 1'b0) begin
            Y = A;
        end
        else begin
            Y = B;
        end
    end
endmodule
```



Simulasi Fungsional dengan GTK Wave No.1



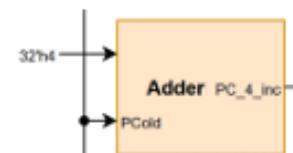
Simulasi Fungsional dengan Quartus No.1



Simulasi Timing dengan Quartus No.1

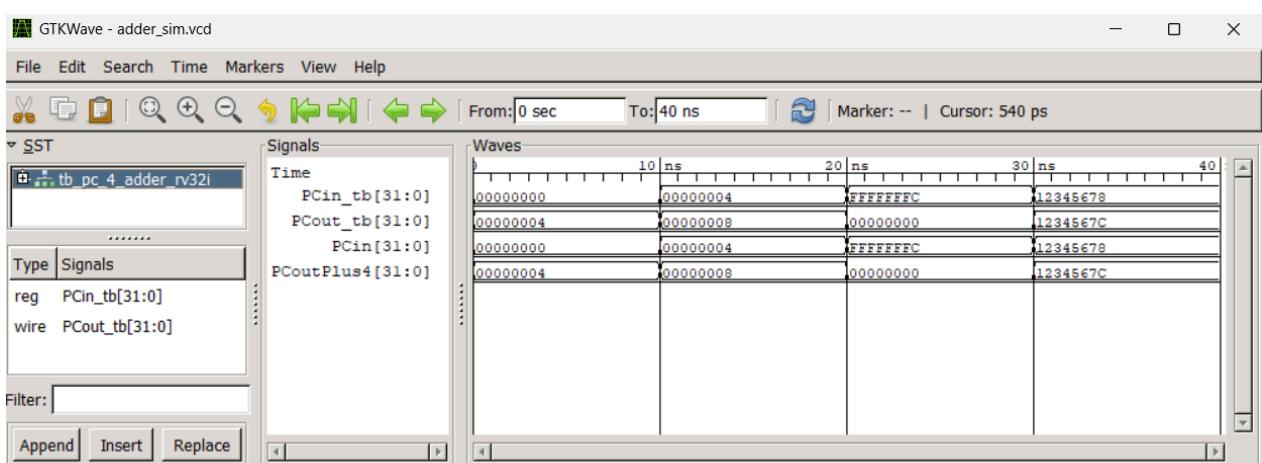
Dengan ini, Mux Synthesizable dan Selector menentukan hasil yang dikeluarkan pada out dan sudah berhasil. Tervalidasi.

- Rancang dan verifikasi komponen **4-adder** selebar 32-bit yang menghitung $PC_{4_inc} = PCold + 4$. Desain memiliki nama `pc_4_adder_rv32i.v`, harus **synthesizable** dan diuji dengan **simulasi fungsional** serta **simulasi timing** (*gate-level* + anotasi penundaan). Cantumkan kode dan tangkapan layar *waveform* dari kedua simulasi.

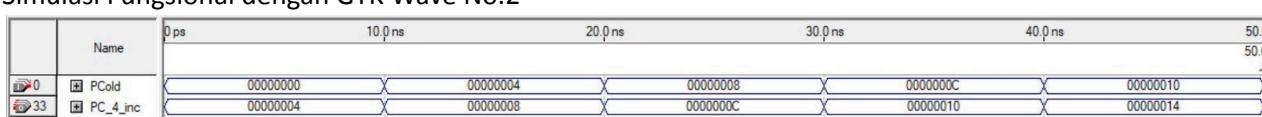


Kode

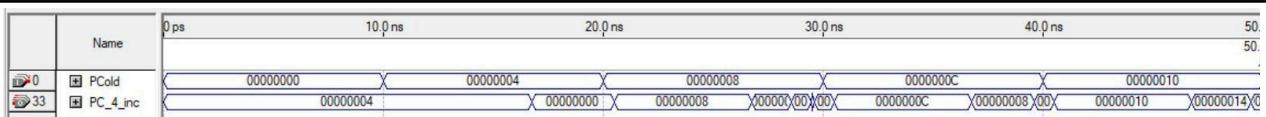
```
// Praktikum EL3011 Arsitektur Sistem Komputer
// Modul      : 2
// Percobaan  : 2
// Tanggal    : 19 November 2025
// Nama (NIM) 1 : William Anthony (13223048)
// Nama (NIM) 2 : Agita Trinanda Ilmi (13223003)
// Nama File   : pc_4_adder_rv32i.v
// Deskripsi   : Desain 4-adder 32-bit untuk perhitungan PCin + 4
module pc_4_adder_rv32i (
    input wire [31:0] PCold,
    output wire [31:0] PC_4_inc
);
    assign PC_4_inc = PCold + 32'd4;
endmodule
```



Simulasi Fungsional dengan GTK Wave No.2



Simulasi Fungsional dengan Quartus No.2



Simulasi Timing dengan Quartus No.2

PCin_tb: Input ke 4-adder (sesuai dengan PCold_tb dari testbench sebelumnya, namun pada waveform terlihat sebagai PCin_tb).

PCout_tb: Output dari 4-adder (sesuai dengan PC_4_inc_tb dari testbench sebelumnya, namun pada waveform terlihat sebagai PCout_tb).

Dari 0 ns hingga sekitar 10 ns. PCin_tb adalah 00000000. PCout_tb adalah 00000004. Ini sesuai dengan harapan, karena $0x00000000 + 4 = 0x00000004$.

Waktu sekitar 10 ns hingga 20 ns. PCin_tb berubah menjadi 00000004.

PCout_tb berubah menjadi 00000008. Ini sesuai dengan harapan, karena $0x00000004 + 4 = 0x00000008$.

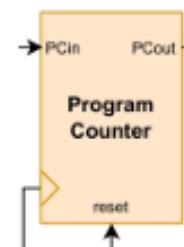
Dari sekitar 20 ns hingga 30 ns. PCin_tb berubah menjadi FFFF_FFFC. PCout_tb berubah menjadi 00000000. Ini sesuai dengan harapan untuk kasus wrap-around (overflow) pada aritmetika 32-bit, karena $0xFFFF_FFFC + 4 = 0x1_0000_0000$, yang terpotong menjadi 0x0000_0000 dalam 32-bit.

Dari sekitar 30 ns hingga 40 ns.

PCin_tb berubah menjadi 12345678. PCout_tb berubah menjadi 1234567C. Ini sesuai dengan harapan, karena $0x12345678 + 4 = 0x1234567C$.

Kesimpulan Fungsional 4-Adder: Waveform ini secara konsisten menunjukkan bahwa PCout_tb selalu menghasilkan nilai PCin_tb ditambah 4 untuk berbagai nilai input, termasuk kasus batas seperti wrap-around. Ini membuktikan bahwa komponen 4-adder 32-bit telah berhasil diimplementasikan secara fungsional, timing, dan synthesizable.

- Buatlah komponen **program counter (PC)** selebar 32-bit untuk arsitektur single-cycle RV32I. PC merupakan register yang akan mengeluarkan dan menahan input berupa nilai PC sendiri setiap **clock rising**, kemudian mengeluarkan nilai 32'h0 saat sinyal **reset falling**. Desain memiliki nama **pc_rv32i.v**, harus **synthesizable** dan diuji dengan **simulasi fungsional** serta **simulasi timing** (gate-level + anotasi penundaan). Cantumkan kode dan tangkapan layar **waveform** dari kedua simulasi.



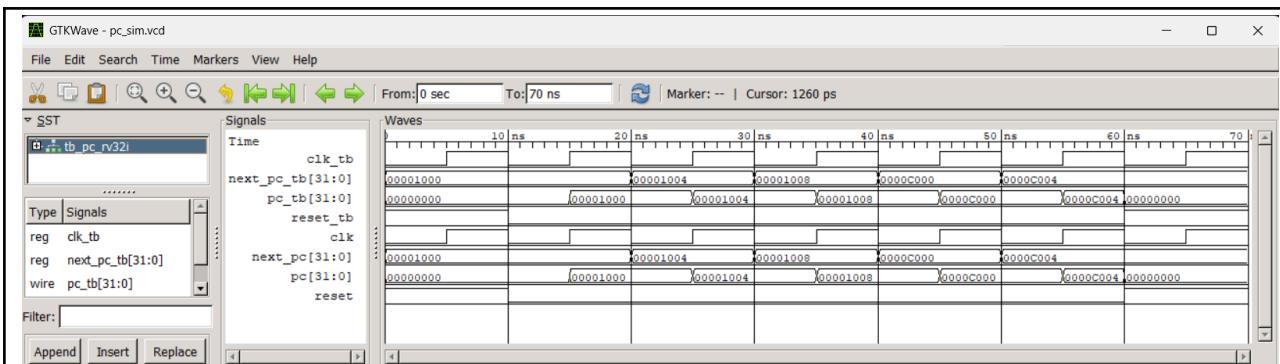
Kode

```

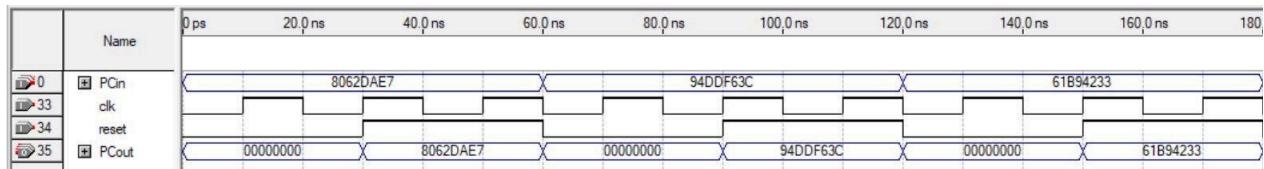
// Praktikum EL3011 Arsitektur Sistem Komputer
// Modul      : 2
// Percobaan  : 3
// Tanggal   : 19 November 2025
// Nama (NIM) 1 : William Anthony (13223048)
// Nama (NIM) 2 : Agita Trinanda Ilmi (13223003)
// Nama File   : pc_rv32i.v
// Deskripsi   : Desain Program Counter (PC) 32-bit untuk RV32I

module pc_rv32i (
    input wire clk,
    input wire reset,
    input wire [31:0] PCin,
    output reg [31:0] PCout
);
    always @ (posedge clk or negedge reset) begin
        if (reset == 1'b0) begin
            PCout <= 32'h0;
        end
        else begin
            PCout <= PCin;
        end
    end
endmodule

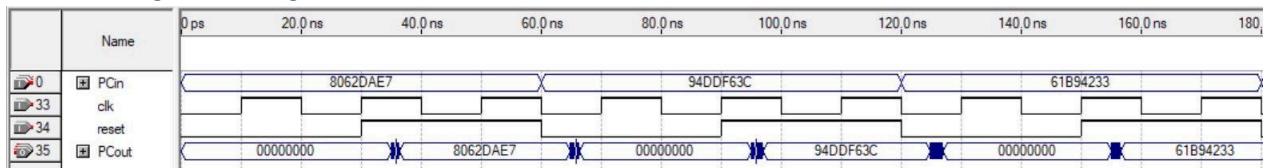
```



Simulasi Fungsional dengan GTK Wave No.3



Simulasi Fungsional dengan Quartus No.3



Simulasi Fungsional dengan Quartus No.3

clk_tb: Sinyal clock.

reset_tb: Sinyal reset (aktif-tinggi asinkron).

next_pc_tb: Input untuk nilai PC selanjutnya.

pc_tb: Output Program Counter saat ini.

Dari 0 ns hingga sekitar 10 ns, reset_tb bernilai 1 (aktif).

pc_tb adalah 00000000. Ini menunjukkan bahwa PC berhasil direset ke 0 saat reset_tb aktif, sesuai dengan desain asinkron reset. Dari sekitar 10 ns hingga 20 ns, reset_tb berubah menjadi 0 (tidak aktif).

Pada tepi naik (posedge) pertama clk_tb setelah reset_tb tidak aktif (sekitar 15 ns):

next_pc_tb adalah 0001000. pc_tb berhasil diperbarui dari 00000000 menjadi 0001000. Ini menunjukkan bahwa PC berhasil memuat next_pc_tb pada tepi naik clock saat reset tidak aktif.

Dari sekitar 20 ns hingga 30 ns. Pada tepi naik clk_tb berikutnya (sekitar 25 ns), next_pc_tb adalah 0001004. pc_tb berhasil diperbarui dari 0001000 menjadi 0001004. Ini menunjukkan perilaku inkrement PC yang benar.

Dari sekitar 30 ns hingga 40 ns.

Pada tepi naik clk_tb berikutnya (sekitar 35 ns), next_pc_tb adalah 0001008.

pc_tb berhasil diperbarui dari 0001004 menjadi 0001008.

Dari sekitar 40 ns hingga 50 ns, next_pc_tb berubah menjadi 000C000 (simulasi lompatan).

Pada tepi naik clk_tb berikutnya (sekitar 45 ns), pc_tb berhasil diperbarui dari 0001008 menjadi 000C000. Ini menunjukkan kemampuan PC untuk memuat alamat target lompatan.

Dari sekitar 50 ns hingga 60 ns, next_pc_tb berubah menjadi 000C004.

Pada tepi naik clk_tb berikutnya (sekitar 55 ns), pc_tb berhasil diperbarui dari 000C000 menjadi 000C004.

Dari sekitar 60 ns hingga 70 ns, reset_tb kembali menjadi 1 (aktif).

Segera setelah reset_tb menjadi 1 (bukan pada tepi clock, karena ini asinkron), pc_tb kembali menjadi 00000000. Ini mengkonfirmasi fungsionalitas reset asinkron. Dengan ini, terbukti kalau komponen Program Counter 32-bit telah berhasil diimplementasikan secara fungsional, timing dan synthesizable.

4. Terjemahkan rentetan instruksi RISC-V (RV32I) berikut menjadi *machine code* dalam format heksadesimal, kemudian tunjukkan/tuliskan **opcode**, **funct7**, **funct3**, dan **trimmed_instr**-nya (mengacu dari subbab [Metodologi Praktikum, Tugas 2 : Immediate Selector](#)), juga dalam bentuk heksadesimal.

```

add x5, x6, x7
sub x10, x10, x11
addi x5, x5, -1
lw x9, 12(x2)
sw x9, -16(x2)
beq x5, x0, -12
jal x1, 80
lui x20, 0x00012

```

Diberikan pada tabel berikut.

9. Tabel referensi untuk tugas 2

No	Instr uksi	Ty pe	Bit [31:25]	Bit [24:20]	Bit [19:15]	Bit [14:12]	Bit [11:7]	Bit [6:0]	Machine code	trimmed _instr
1	add x5 x6 x7	R	funct7	rs2	rs1	funct3	rd	opcode	0x00730 2b3	0x00E60 5
			0x00	0x07	0x06	0x00	0x05	0x33		
			000 0000	0 0111	0 0110	000	0 0101	011 0011		
2	sub x10 x10 x11	R	funct7	rs2	rs1	funct3	rd	opcode	0x40B50 533	0x816A0 A
			0x20	0x0B	0xA	0x0	0xA	0x33		
			010 0000	0 1011	0 1010	000	0 1010	011 0011		
3	addi x5 x5 -1	I	imm		rs1	funct3	rd	opcode	0xFFFF28 293	0x1FFE5 05
			0x0FFF		0x05	0x0	0x05	0x13		
			000 1111 1111 111		00101	000	0 0101	001 0011		
4	lw x9 12(x2)	I	imm		rs1	funct3	rd	opcode	0x00C12 483	0x00182 49
			0x0000C		0x02	0x2	0x09	0x03		
			0000 0000 0000 1100		00010	010	0 1001	000 0011		
5	sw x9 -16(x 2)	S	imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	0xFE912 823	0x1FD22 50
			0xFF0 [11:5]	0x09	0x02	0x2	0xFF0 [4:0]	0x23		
			111 1111	01001	00010	010	1 0000	010 0011		
6	beq x5 x0 -12	B	imm[12][imm1 0:5]	rs2	rs1	funct3	imm[4:1] imm[11]	opcode	0xFE028 AE3	0x1FC05 15
			0xFF4 [12] 0xFF4[10:5]	0x00	0x05	0x0	0xFF4 [4:1] 0xFF4[11]	0x63		
			1 11 1111	00000	00101	000	1 0101	110 0011		
7	jal x1 80	J	imm[20] imm[10:5]	imm[4:1]im m[11]	imm[19: 15]	imm[14: 12]	rd	opcode	0x05000 0EF	0x00A00 01
			0x50[20] 0x50[10:5]	0x50[4:1] 0x50[11]	0x50[19: 15]	0x50[14: 12]	0x01	0x6F		
			0 00 0101	0 0000 1	0 0000	000	0 0001	110 1111		
8	lui x20 0x00 012	U	imm[31:25]	imm[24:20]	imm[19: 15]	imm[14: 12]	rd	opcode	0x00012 A37	0x00002 54
			0x12 [31:25]	0x12 [24:20]	0x12 [19:15]	0x12 [14:12]	0x14	0x37		
			000 0000	0 0000	00010	010	1 0100	011 0111		

Mengikuti pada format RISC-V, RV32I memiliki lebar instruksi 32-bit. Format atau tipe dasarnya ada enam: R, I, S, B, U, J.

Format	Struktur Bit	Keterangan & Contoh
R-type	`funct7 [31:25]	rs2 [24:20]
I-type	`imm[11:0]	rs1
S-type	`imm[11:5]	rs2
B-type	`imm[12]	imm[10:5]
U-type	`imm[31:12]	rd
J-type	`imm[20]	imm[10:1]

Dengan ini komponen dari format dasar instruksi dijelaskan pada tabel selanjutnya.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																opcode
I																															opcode	
S																															opcode	
B																															opcode	
U																															opcode	
J																															opcode	

Untuk keterangan masing-masing *field*, ada di sini :

Komponen	Keterangan
opcode [6:0]	Menentukan kelas/format instruksi dan keluarga operasinya.
rd/rs1/rs2	Indeks register tujuan (rd) dan sumber (rs1, rs2).
funct3/funct7	Sub-opcode untuk membedakan varian operasi dalam satu opcode.
immediate	Konstanta bertanda; tata letak tergantung format (PC-relative untuk B/J, upper-immediate untuk U).

5. Dengan mengacu pada *machine code* yang kalian sudah dapatkan di nomor sebelumnya, tentukan seluruh keluaran CU (16 sinyal). Buatlah dalam bentuk tabel, dengan instruksi dibuat per baris dan sinyal dibuat per kolom, seperti pada subbab [Landasan Teoretis Praktikum, Control Unit \(CU\)](#).

Keluaran berdasarkan Field dimasukkan dalam kode diberikan sebagi berikut.

8. Tabel untuk Tugas 1

No	Instruksi	Machine code	Sinyal Control Unit															
			cu_ALU1src	cu_ALU2src	cu_immtyp e	cu_ALUtyp e	cu_adtyp e	cu_gatyp e	cu_shiftyp e	cu_slype	cu_rdtyp e	cu_rdwrtyp e	cu_loadtyp e	cu_stor e	cu_storetyp e	cu_bra nch	cu_bran chtyp e	cu_Pcty pe
1	add x5 x6 x7	0x007302b3	0	0	---	00	0	--	--	-	00	1	---	0	--	0	---	0
2	sub x10 x10 x11	0x40b50533	0	0	---	00	1	--	--	-	00	1	---	0	--	0	---	0
3	addi x5 x5 -1	0xffff28293	0	1	000	00	0	--	--	-	00	1	---	0	--	0	---	0
4	lw x9 12(x2)	0x00c12483	0	1	000	00	0	--	--	-	01	1	010	0	--	0	---	0
5	sw x9 -16(x2)	0xfe912823	0	1	001	00	0	--	--	-	--	0	---	1	10	0	---	0
6	beq x5 x0 -12	0xfe028ae3	1	1	011	00	0	--	--	-	--	0	---	0	--	1	000	0
7	jal x1 80	0x050000ef	1	1	100	00	0	--	--	-	10	1	---	0	--	0	---	1
8	lui x20 0x00012a3	0x00012a37	0	1	011	00	0	--	--	-	11	1	---	0	--	0	---	0

Secara rincinya, dibagi atas 3 format yaitu Sinyal ALU & Immediate, Sinyal Writeback & Load/Store (Memory), dan terakhir Sinyal Branch & PC. Sinyal ALU & Immediate untuk perhitungan, Sinyal Writeback & Load/Store untuk Penyimpanan isi Memory, dan Sinyal Branch & PC untuk arah aliran program (Datapath). Dirincikan dengan tabel sebagai berikut.

Tabel Pertama, Sinyal ALU & Immediate

Instruksi	ALU1src	ALU2src	immtyp e	ALUtyp e	adtyp e	gatyp e	shiftyp e	slype
ADDI	0	1	0	0	0	0	0	0
BEQ	1	1	10	0	0	0	0	0
LW	0	1	0	0	0	0	0	0

Tabel Kedua, Sinyal Writeback & Memory

Instruksi	rdtyp e	rdwrite	loadtyp e	store	storetyp e
ADDI	0	1	10	0	0
BEQ	0	0	10	0	0
LW	1	1	10	0	0

Tabel Ketiga, Sinyal Branch & PC

Instruksi	branch	branchtype	PCtype
ADDI	0	0	0
BEQ	1	0	1
LW	0	0	0

Daftar Referensi

Patterson, David, dan John Hennessy. Computer Organization and Design : The Hardware/Software Interface. 2012. Waltham : Elsevier Inc.

D. Harris & S. Harris, Digital Design and Computer Architecture: RISC-V Edition.

Sutandi, Stefen, dkk. M1 rev.2 Modul Praktikum EL3011 Arsitektur Sistem Komputer Edisi 2025 untuk digunakan pada Semester I Tahun Akademik 2025 / 2026. 2025. Ms Teams (Diakses 18 November 2025).

Sutandi, Stefen, dkk. Modul 2 Praktikum EL3011 Arsitektur Sistem Komputer Edisi 2025 SYNTHESIZABLE RISC-V (RV32I) MICROPROCESSOR BAGIAN II : CONTROL UNIT (CU), IMMEDIATE SELECTOR, BRANCHER, PROGRAM COUNTER (PC), 4-ADDER, DAN 2-TO-1 MUX GENERIK Semester I Tahun Akademik 2025 / 2026. 2025. Ms Teams (Diakses 18 November 2025).

Venus. <https://github.com/61c-teach/venus/wiki>. (Diakses 18 November 2025)