



## Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Arsitektur Sistem Komputer (EL3111)  
Tahun / Semester : 2023-2024 / Ganjil  
Modul : 5-Synthesizable MIPSpc32® Microprocessor Bagian III  
Top Level Design Dan Testbench  
Nama Asisten / NIM : Emmanuella Pramudita / 13220031  
Nama Praktikan / NIM : Dimas Ridhwana Shalsareza / 13221076

### Tugas Pendahuluan

1. Diberikan sebuah kode program dalam bahasa C sebagai berikut

- a. Ubahlah kode dalam bahasa C tersebut ke dalam bahasa assembly MIPS32® menggunakan instruksi yang dapat dimengerti oleh mikroprosesor Single-Cycle MIPS32® yang diimplementasikan dalam praktikum ini. Jangan lupa untuk menggunakan instruksi nop setelah instruksi yang berkaitan dengan branching dilaksanakan.

```
.data
    sum: .word 0

.text
.globl main

main:
    li $s0, 0
    li $s1, 0

loop:
    beq $s1, 10, end_loop

    addi $s0, $s0, 1    # sum = sum + 1
    addi $s1, $s1, 1    # i = i + 1
    nop

    j loop
    nop

end_loop:
    # Code after the loop ends (if any)
```

- b. Ubahlah program bahasa assembly yang telah dibuat pada (a) menjadi file objek yang berisi urutan bilangan biner untuk masing-masing instruksi. Contohnya, apabila terdapat instruksi dalam bahasa assembly `addi $s1, $s1, 1`, maka kode biner yang bersesuaian adalah `0x22310001`.

```
0x34100000
0x34110000
0x3401000a
0x10310006
0x22100001
0x22310001
0x00000000
0x08100002
0x00000000
```

**2. Diberikan urutan instruksi dalam bahasa assembly MIPS32® sebagai berikut.**

**a. Jelaskan maksud dari bahasa assembly tersebut.**

Secara umum, bahasa assembly tersebut melakukan beberapa operasi aritmatika, manipulasi memori (penyimpanan dan pengambilan), dan mengakhiri program dengan syscall untuk keluar.

**b. Ubahlah program bahasa assembly menjadi file objek yang berisi kode biner yang merepresentasikan masing-masing instruksi.**

```
0x20100013
0x20110015
0x16530003
0x00000000
0x08100005
0x02119822
0x22730000
0x22140004
0xae910000
0x8e950000
0x02a0a820
0x0810000d
0x00000000
0x3402000a
0x0000000c
```

**3. Apa yang dimaksud datapath? Bagaimana hubungan antara datapath dengan control?**

Datapath merupakan komponen dari mikroprosesor yang melakukan operasi aritmetik serta melakukan penyimpanan data. Terdapat lima tahap pemrosesan instruksi meliputi instruction fetch, instruction decode, execute, memory access, dan write back. Sedangkan control merupakan komponen dari mikroprosesor yang mengatur datapath berdasarkan instruksi yang sedang dieksekusi. Maka hubungan control dengan datapath terlihat sesuai dengan sifat control tersebut.

**4. Bagaimana datapath eksekusi instruksi tersebut pada mikroprosesor Single-Cycle MIPS32®?**

```
ALUOut RegisterFile (0x00000000) => ALUIn1 ALU,
ReadData2 RegisterFile (0x00000013) => ALUIn2 ALU,
PC Register (0x00000000) => Address Instruction Memory,
Instruction Instruction Memory (0x20100013) => Instruction Register,
ReadData1 RegisterFile (0x00000000) => WriteData RegisterFile,
ALUOut ALU (0x00000013) => WriteData Data Memory,
ReadData1 RegisterFile (0x00000013) => ALUIn1 Data Memory,
ALUOut ALU (0x00000000) => ALUIn2 Data Memory,
ALUOut Data Memory (0x00000000) => ReadData RegisterFile,
```

5. Gambarkan dalam bentuk tabel isi instruction memory, data memory, dan register file pada akhir program yang dibuat pada soal 1 dan 2

Data memory	Instruction Memory	Register	Data memory	Instruction Memory	Register
0x00400000	0x20100013	\$16	0x00400000	0x34100000	\$s0
0x00400004	0x20110015	\$17	0x00400004	0x34110000	\$s1
0x00400008	0x16530003	\$18	0x00400008	0x3401000a	\$s1
0x0040000C	0x00000000		0x0040000C	0x10310006	
0x00400010	0x08100005		0x00400010	0x22100001	\$s0
0x00400014	0x02119822	\$19	0x00400014	0x22310001	\$s1
0x00400018	0x22730000	\$19	0x00400018	0x00000000	
0x0040001C	0x22140004	\$20	0x0040001C	0x08100002	
0x00400020	0xae910000	\$17	0x00400020	0x00000000	
0x00400024	0x8e950000	\$21			