

Tugas Mandiri 2

Deadline: **24** Maret 2023, 17.00 Waktu SCELE

Pengantar Organisasi Komputer
Semester Genap 2022/2023

Revisi 0

Petunjuk pengerjaan:

1. Jawaban tugas **HARUS** ditulis tangan atau menggunakan pen tablet. Untuk penulisan tangan dapat ditulis di kertas polos A4 atau kertas folio bergaris dan discan. **Tidak boleh diketik!** Kumpulkan semua halaman jawaban ke satu file yang berformat PDF.
2. Tuliskan **nama, NPM, kelas, dan kode asisten** di **setiap lembar** berkas jawaban Anda. Kelalaian menuliskan keempat informasi ini (lupa atau salah tulis) akan **diberikan penalti -5 poin/kesalahan**.
3. Pastikan tulisan tangan rapi dapat dibaca!
4. Keterlambatan **10 menit < x < 2 jam** dari deadline akan dikenakan penalti **sebesar 30 poin** dari nilai tugas. Keterlambatan **2-6 jam** dari deadline dikenakan penalti **sebesar 60 poin** dari nilai tugas. Pengumpulan yang melewati batasan waktu tersebut **tidak akan dinilai**.
5. Silahkan tambahkan asumsi sendiri bila diperlukan.
6. **Plagiarisme adalah pelanggaran serius dengan sanksi nilai 0.**
7. **Warna biru** menunjukan revisi, silahkan tulis pada revisi mana anda mengerjakan, jika tidak menulis maka akan dianggap revisi terbaru.
8. Format penamaan:

Tugas2_KodeAsdos_NPM_Nama.pdf

Contoh: **Tugas2_RT_2206110311_Suzumelwato.pdf**

Revisi 0

-
1. **[15]** Peokra, yang sedang mempelajari MIPS Instruction Set Architecture, merasa kesulitan dalam memahami format-format instruksi yang terdapat di MIPS ISA, yakni R-Format, I-Format, dan J-Format. Sebagai seorang teman yang baik, Peokra meminta bantuan kepadamu untuk menjelaskan beberapa hal mengenai perbedaan dari ketiga format instruksi tersebut. Apakah perbedaan dari ketiga format instruksi tersebut? Bagaimana ciri umum mereka terhadap alur Datapath (termasuk control unit)? (misalkan, format tertentu akan menyebabkan nilai 0 atau 1 pada bagian tertentu)

PERBEDAAN	R-FORMAT	I-FORMAT	J-FORMAT
BASIC INSTRUCTION FORMAT	Opcode : 6 bit rs (register source) : 5 bit rt (register target) : 5 bit rd (register destination) : 5 bit shamt : 5 bit funct : 6 bit	Opcode : 6 bit rs (register source) : 5 bit rt (register target) : 5 bit immediate : 16 bit	Opcode : 6 bit target address : 26 bit
DATA PATH (CONTROL UNIT)	RegDest : 1 ALUSrc : 0 MemtoReg : 0 RegWrite : 1 MemRead : 0 MemWrite : 0 Branch : 0 ALUOp1 : 1 ALUOp2 : 0 Jump : 0	lw sw beq RegDest : 0 X X ALUSrc : 1 1 0 MemtoReg : 1 X X RegWrite : 1 0 0 MemRead : 1 0 0 MemWrite : 0 1 0 Branch : 0 0 1 ALUOp1 : 0 0 0 ALUOp2 : 0 0 1 Jump : 0 0 0	RegDest : X ALUSrc : X MemtoReg : X RegWrite : X MemRead : X MemWrite : X Branch : X ALUOp1 : X ALUOp2 : X Jump : 1

2. **[4+4+4+4+4]** Peokra sedang membuat suatu program menggunakan MIPS assembly. Namun, terdapat beberapa instruksi dalam programnya yang diragukan kebenarannya. Periksa apakah instruksi-instruksi di bawah ini dapat dijalankan atau tidak! Jika iya, sebutkan fungsi/kegunaan dari instruksi tersebut beserta register yang terdampak (isinya berubah)! Jika tidak, cukup berikan alasan mengapa instruksi tersebut tidak dapat dijalankan!

- a. `addi $9, $8, 0xAAAAA`

TIDAK DAPAT DIJALANKAN

Instruksi tersebut memiliki format yang salah karena nilai immediate lebih dari 16 bit.

- b. `jal myLabel`

DAPAT DIJALANKAN

Jump and Link digunakan untuk memanggil suatu prosedur (myLabel) dan menyimpan alamat instruksi yang akan dilanjutkan setelah prosedur tersebut selesai dieksekusi

Register yang terdampak : \$31 atau \$ra (return address)

- c. `sw $8, $10($a0)`

TIDAK DAPAT DIJALANKAN

Instruksi tersebut memiliki format yang salah karena nilai immediate diisi dengan register (\$10)

d. sub \$v2, \$a0, \$a1

TIDAK DAPAT DIJALANKAN

Tidak terdapat register \$v2

e. lw \$a0, -4(\$t0)

DAPAT DIJALANKAN

Load word digunakan untuk mengakses memori dengan alamat offset (rs) (-4(\$t0)) dan menyimpannya pada rt (\$a0)

Register yang terdampak : \$t0 atau \$a0 (arguments)

3. **[4+4+4+4+4]** Peokra menemukan beberapa bilangan yang merepresentasikan instruksi MIPS. Tentukanlah tipe format serta instruksi MIPS apa yang direpresentasikan oleh bilangan-bilangan dibawah ini! Tuliskan cara mendapatkan instruksi tersebut! Sebutkan ciri khusus yang dapat digunakan untuk mengidentifikasi/membedakan masing-masing instruksi tersebut!

Untuk membedakan masing 2 instruksi,
kita bisa lihat dari opcode nya

Opcode	Format
0 _{hex}	R
2 _{hex} / 3 _{hex}	J
lainnya	I

a. (0001 0101 0100 1001 1111 1111 1111 1011)₂

opcode	rs	rt	immediate
000101	01010	01001	111111111111011
bne	\$t2	\$t1	0xFFFFB

Instruksi : **bne \$t2, \$t1, 0xFFFFB**

Tipe : **I-format**

b. $(02082020)_{16}$

$(0000\ 0010\ 0000\ 1000\ 0010\ 0000\ 0010\ 0000)_2$

opcode	rs	rt	rd	shamt	function
000000	10000	01000	00100	00000	100000
r-format	\$s0	\$t0	\$a0	0	add

Instruksi : **add \$a0, \$s0, \$t0**
 Tipe : **R-format**

c. $(1515776412)_8$

$(001\ 101\ 001\ 101\ 111\ 111\ 110\ 100\ 001\ 010)_2$
 $= (0000\ 1101\ 0011\ 0111\ 1101\ 0000\ 1010)_2$

opcode	target address
000011	01001101111110100001010
jal	0x137F00A

Instruksi : **jal 0x137F00A**
 Tipe : **J-format**

d. $(600256)_{10}$

$(0000\ 0000\ 0000\ 1001\ 0010\ 1000\ 1100\ 0000)_2$

opcode	rs	rt	rd	shamt	function
000000	00000	01001	00101	00011	00000
r-format	\$zero	\$t1	\$a1	3	sll

Instruksi : **sll \$a1, \$t1, 3**
 Tipe : **R-format**

e. $(ACABAAA9)_{16}$

$(1010\ 1100\ 1010\ 1011\ 1010\ 1010\ 1010\ 1001)_2$

opcode	rs	rt	immediate
101011	00101	01011	10101010101001
sw	\$a1	\$t3	0xAAA9

Instruksi : **sw \$t3, 0xAAA9(\$a1)**
 Tipe : **I-format**

Reminder: Jika bilangan di atas tidak berukuran 32 bit, extend terlebih dahulu MSB-nya dengan bit 0 hingga memenuhi 32 bit.

4. **[20]** Setelah mempelajari MIPS ISA, Peokra ingin mencoba untuk membuat suatu program untuk membantunya dalam pekerjaannya. Peokra ditugaskan untuk membuat suatu program yang dapat menghitung suku-suku yang terdapat dalam barisan $(3, 9, 27, 81, 243, \dots, 3^{n-1}, 3^n)$ hingga suku ke- n . Ia pun meminta bantuan kepadamu untuk membuat suatu program yang dapat melakukan hal itu, dengan spesifikasi sebagai berikut:

- Nilai n diketahui sudah tersimpan di dalam memori.
- Dijamin nilai n merupakan integer dalam interval $1 \leq n \leq 18$
- Suku-suku dari barisan tersebut perlu disimpan pada memori pada alamat yang berurutan (seperti misalkan `0x10011200`, `0x10011204`, `0x10011208`, dst).
- Pada awalnya, register `$a0` berisi address memori yang menyimpan nilai n , dan register `$a1` berisi address memori yang akan digunakan untuk menyimpan suku pertama barisan tersebut.

Buatlah program yang memenuhi permintaan Peokra di atas menggunakan MIPS assembly!

```

.text
.globl main
main:
#----- MENGAMBIL NILAI n DARI MEMORI -----#
    lw $a0, 0($a0)
    # $a0 = Memory address yang menyimpan nilai n

#----- MENYIMPAN ELEMEN PERTAMA DI MEMORI -----#
    li $t0, 3                # Inisialisasi $t0 = 3
    sw $t0, 0($a1)
    # $a1 = Memory address yang menyimpan suku pertama barisan

    # Cek jika n-1 == 0, maka langsung akhiri perhitungan
    subi $a0, $a0, 1
    beqz $a0, selesai

#----- PERHITUNGAN 3^n -----#
loop:
    addi $t1, $t1, 1        # $t1 = counter loop

    mult $t0, $t0
    mflo $t0                # $t0 = $t0^2
                           # Mengambil hasil perkalian

    addi $a1, $a1, 4        # Pindah ke word selanjutnya
    sw $t0, 0($a1)         # Simpan ke memori

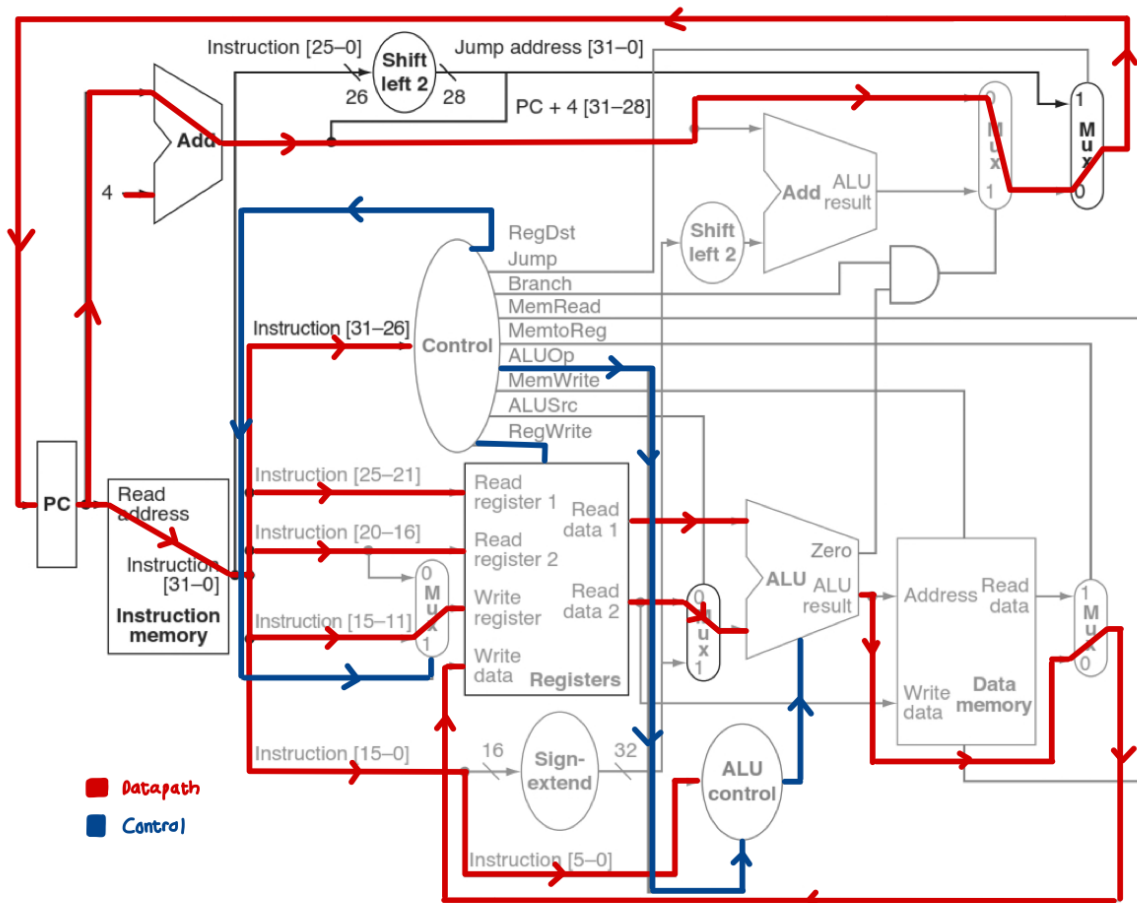
    subi $a0, $a0, 1
    bnez $a0, loop

#----- MENGAKHIRI PROGRAM -----#
selesai:
    li $v0, 10
    syscall

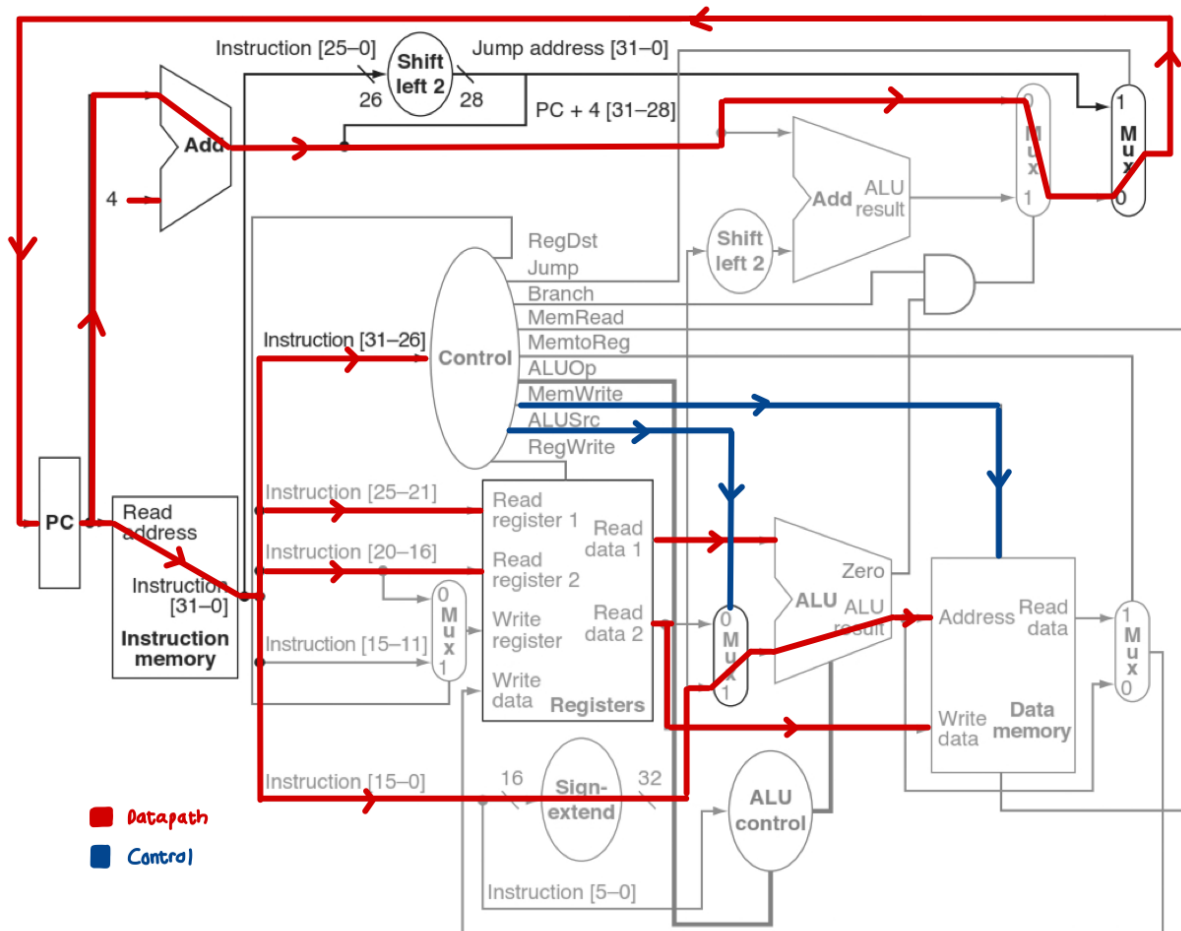
```

5. **[5+5+5]** Setelah memahami MIPS ISA (berkat bantuan kamu!), sekarang saatnya Peokra mempelajari tentang Datapath dan Control. Namun, Peokra kembali mengalami kebingungan untuk menentukan alur Datapath dan Control saat suatu instruksi MIPS dijalankan. Oleh karena itu, Peokra meminta bantuan kalian untuk menggambarkan jalur yang dilewati datapath beserta control saat menjalankan instruksi-instruksi berikut:

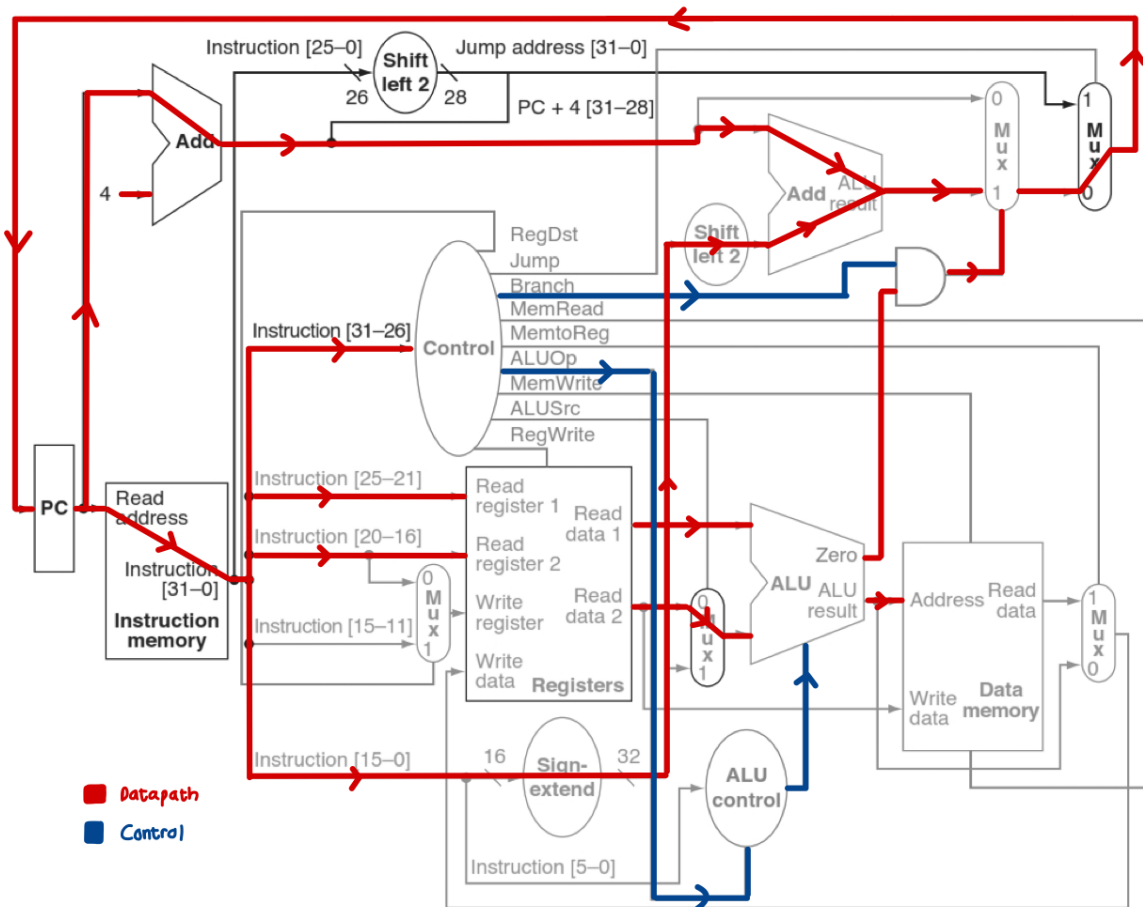
a. add \$16, \$10, \$12



b. sw \$9, 8(\$19)



c. beq \$18, \$14, someLoop (asumsikan nilai \$18 == nilai \$14 sehingga branching akan terjadi)



Ketentuan menjawab:

- Tandai alur datapath dengan warna merah.
- Tandai alur control signal yang pasti memiliki nilai 1 saat menjalankan suatu instruksi dengan warna biru. Khusus untuk control signal `ALUOp`, tandai control signal tersebut apabila memiliki nilai selain 0.

Note:

Untuk mengerjakan soal ini, gunakanlah gambar datapath yang disediakan pada link [MIPS Datapath \(r0\).png](#). Gambar dapat diperbanyak secara digital atau dicetak fisik (print). Untuk soal ini, diperbolehkan menjawab dengan mengedit gambar menggunakan aplikasi digital (seperti misalnya *Paint*, *GIMP*, *Photoshop*, dll.). Terdapat 3 buah gambar datapath yang harus dikumpulkan (satu untuk masing-masing instruksi).

6. **[10]** Isilah tabel berikut sesuai dengan kode assembly nomor 5a berdasarkan nilai yang dilewati pada datapath.

RegDst	1
RegWrite	1
WriteReg (binary, 5 bit)	(10000) ₂
ReadReg1 (binary, 5 bit)	(01010) ₂
ReadReg2 (binary, 5 bit)	(01100) ₂
ReadData1 (hex, 32 bit)	0x 03040CA0
ReadData2 (hex, 32 bit)	0x 00201020
ALUSrc	0
ALU Input 2 (hex, 32 bit)	0x 00201020
ALU Control (binary, 4 bit)	00 10
ALU Result (hex, 32 bit)	0x 032429C0
Zero	0
MemWrite Data (hex, 32 bit)	0x 00201020
MemWrite	0
MemRead	0
MemToReg	0

Note: dalam mengisi tabel di atas, gunakan nilai-nilai dari tabel ini sebagai referensi:

Registers in MIPS	Register initial values	ALU Control (4 bit binary)
R2-R3 = \$v0-\$v1 R4-R7 = \$a0-\$a3 R8-R15 = \$t0-\$t7 R16-R23 = \$s0-\$s7	R10 = 0x 0304 0CA0 R11 = 0x 0000 0B16 R12 = 0x 0020 1D20 R16 = 0x 0043 C27A R17 = 0x 0002 0A11	AND = 0000 OR = 0001 ADD = 0010 SUB = 0110 SLT = 0111 NOR = 1100