**Simulasi Prosesor, dan Sistem Komputer**

Tujuan pembuatan: Membuat simulator system computer hardware (stored-program computer) dalam bentuk software yang meliputi minimal prosesor (ALU), control unit (CU), main memori (RAM), system computer (integrasi ALU, control unit, dan main memori) sebagai sarana bantu untuk menjelaskan dan media pembelajaran konsep dasar rancang sistem operasi.

Sasaran yang ingin dicapai: Simulator dapat digunakan untuk memfasilitasi poin 1, dengan didukung bila poin 2, dan poin 3 telah dipenuhi

1. Menerapkan dan mengimplementasi konsep dan algoritma-algoritma khususnya mengenai manajemen proses, manajemen memori, dan deadlock.
2. Memberikan ruang latihan membuat code generator, yaitu membuat file yang berisi program yang dapat dieksekusi oleh simulator.
3. Memberikan ruang latihan membuat compiler, dengan bahasa program yang dirancang sendiri, dan dapat dijalankan di mesin simulator.

Bahasa pengembangan: Versi pertama, menggunakan bahasa Java, dan berpotensi dikembangkan dalam bahasa C/C++.

Spesifikasi:

Word data: 32 bits

Half word (high, low): 16 bits

Byte: 8 bits

Instruction set size: 32 bits (RISC-like)

Operation code: 8 bits

Banyaknya register: 16 (sementara)

Ukuran register: 32 bits

Immediate data disimpan di lokasi memory: alamat near (16 bit offset)

Alamat: near (16 bit), dan far (32 bit)

Prinsip membaca instruksi: 1 word.

Non-pipelining, instruksi baru (berikutnya) dikerjakan bilamana instruksi yang saat ini dikerjakan telah selesai.

Registers:

* General purpose: r0 hingga r15
* Khusus: Program Counter (PC), Base register (BREG), Limit register (LREG), Flag register (FLAG), Stack pointer (SP), Base pointer (BP), Page register (PAGE), Offset register (OFFSET), Thread register (THREAD), Page fault register (PFAULT),

FLAG (flag register), status diperbaharui setiap kali suatu instruksi dieksekusi

|  |  |  |  |
| --- | --- | --- | --- |
| Bit | Deskripsi | Bit |  |
| 0 | ZF (zero) – 0 (bukan nol), 1 (nol) | 16 |  |
| 1 | SF (sign) – 0 (positif), 1 (negative) | 17 |  |
| 2 | CF (carry) – 0 (no carry), 1 (carry) | 18 |  |
| 3 | OF (overflow) – 0 (no overflow), 1 (overflow) | 19 |  |
| 4 | IF (interrupt) – 0 (enable interrupt), 1 (disable interrupt) | 20 |  |
| 5 | TF (trap) – 0 (no trap), 1 (trap) | 21 |  |
| 6 | PF (parity) – 0 (even), 1 (odd) | 22 |  |
| 7 | GF (page fault) - 0 (no page fault), 1 (page fault) | 23 |  |
| 8 | MF (miss cache) – 0 (hit cache), 1 (miss cache) | 24 |  |
| 9 | DV (divide overflow) – 0 (no divide overflow), 1 – (divide overflow) | 25 |  |
| 10 | XF (execution phase) – 0 (halt), 1 – (execution) | 26 |  |
| 11 |  | 27 |  |
| 12 |  | 28 |  |
| 13 |  | 29 |  |
| 14 |  | 30 |  |
| 15 |  | 31 |  |

Instruction set architectur group :

Head-id (2 bit) = 00

|  |  |  |
| --- | --- | --- |
| Op.code | Deskripsi, syntax | Penjelasan dan status |
| ADD | ADD R[rd], R[rs], R[rt]  Add register ke register | Operasi di register, data di memori, perlu dipindahkan ke register. |
| ADDI | ADDI R[t], R[rs], immediate  Add immediate data ke register | Add immediate (16 bits maksimum) |
| DIV |  | Belum diimplementasi |
| MUL |  | Belum diimplementasi |
| SUB | Subtract inter registers |  |
| SUBI | Subtract immediate data |  |
| JMP | Unconditional jump, ke immediate address | Near |
| JMPR | Unconditional jump, ke address oleh register | far |
| JE | Jump if equal |  |
| JNE | Jump if not equal |  |
| JGT | Jump if greater than |  |
| JLT | Jump if less than |  |
| MOVR | Move register ke register 32 bits (word) | (word, 32 bits) |
| MOV | Move word 32 bits dari register ke memori |  |
| MOVI | Move immediate 16 bit ke register 32 bits |  |
| MOVM | Move word 32 bits dari memori ke register |  |
| MOVB | Move byte dari register ke memori | (data, 8 bits) |
| MOVMB | Move byte memori ke register |  |
| MOVH | Move half word (16 bits) high dari register ke memori (numeric) | (data, 16 bits), belum diimplementasi |
| MOVMH | Move half word (16 bits) high dari memori ke register (numeric) | (data, 16 bits), belum diimplementasi |
| MOVL | Move half word (16 bits) low dari register ke memori (numeric) | (data, 16 bits), belum diimplementasi |
| MOVML | Move half word (16 bits) low dari memori ke register (numeric) | (data, 16 bits), belum diimplementasi |
| MOVMHW | Move 16 bits data dari memori ke register |  |
| MOVD | Move double word dari register ke memori (numeric) | Belum diimplementasi |
| MOVMD | Move double word dari memori ke register (numeric) | Belum diimplementasi |
| MOVC | Move conditional | True (1), false (0), belum diimplementasi |
| CALL | Call procedure | Far atau near, belum diimplementasi |
| RET | Return from procedure | Near, belum dimplementasi |
| RETF | Return from procedure | Far, belum diimplementasi |
| INT | Call interrupt | belum diimplementasi |
| IRET | Return from interrupt | belum diimplementasi |
| PUSH | Push word |  |
| POP | Pop word |  |
| PUSHB |  | ? perlu |
| POPB |  | ? perlu |
| PUSHF | Push flag register to stack | belum diimplementasi |
| POPF | Pop to flag register | belum diimplementasi |
| HLT | Halt system |  |
| CLI | Clear interrupt | belum diimplementasi |
| DLI | Disable interrupt | belum diimplementasi |
| IN | Read from input port | belum diimplementasi |
| OUT | Write to output port | belum diimplementasi |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Format instruksi (32 bytes)

Halt/end program (HALT)

Kode: 00010100 (0x7F)

|  |  |
| --- | --- |
| Op.code |  |
| 8 bits | 24 bits |
| Bit: 31-24 | 0-23 |
| Flag | XF, 0 (jika sudah halt/terminate), dan 1 (jika eksekusi masih berlangsung) |

Arithmetic register ke register (ADD), contoh program perlu direvisi

Kode: 00000000 (0x00), rd = rs1 + rs2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Op.code | Reg destination | Reg. source-1 | Reg. source-2 |  |
| 8 bits | 4 bits | 4 bits | 4 bits | 12 bits |
| Bit: 31-24 | 20-23 | 16-19 | 15-12 | 0-12 |
| Flag | SF, ZF, CF, OF | | | |

Arithmetic immediate (ADDI), contoh program perlu direvisi

Kode: 00000001 (0x01), rd = rs1 + immediate

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. source-1 | Immediate data |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | SF, ZF, CF, OF | | |

Transfer register ke register (MOVR), 32 bits (word), contoh program perlu direvisi

Kode: 00000002 (0x02), rd = rs1

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. source-1 |  |
| 8 bits | 4 bits | 4 bits | 16 bits diabaikan |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | SF, ZF, CF, OF | | |

Transfer immediate data (constant ?16 bit) ke register (MOVI), contoh program perlu direvisi

Kode opcode: 00000011 (0x03), rd = imm

|  |  |  |
| --- | --- | --- |
| Op.code | Reg. destination | Immediate/address |
| 8 bits | 4 bits | 20 bits |
| Bit: 31-24 | 20-23 | 0-19 |
| Flag | SF, ZF, CF, OF | |

Transfer register ke memory (MOV), contoh program perlu direvisi

Kode: 00000100 (0x04), M[rd+offset address] = s1

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination (alamat memori) | Reg. Source-1 (sumber) | Immediate/offset address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag |  | | |

Transfer word dari memori ke register (MOVM)

Kode: 00000101 (0x05), rd = M[s1 + immediate]

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. Source-1 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Format berikut di bawah ini perlu di tata ulang

Transfer byte 8 bits register ke memori (MOVB)

Kode: 00000110 (0x06), M[rd+immediate] = s1

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination (alamat memori) | Reg. source-1 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19[16-17] | 0-15 |
| Flag | Belum selesai | | |

Transfer byte 8 bits dari memori ke register (MOVMB)

Kode: 00000111 (0x07), rd = M[s1+immediate]

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. source-1 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Transfer half word high 16 bits register ke memori (MOVH), 2 byte lokasi

Kode: 00001000 (0x08), M[rd+immediate] = s1:[16-31]

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. source-1 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Transfer half word high 16 bits dari memori ke register (MOVMH)

Kode: 00001001 (0x09)

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. destinasi | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Transfer half word low 16 bits register ke memori (MOVL) [0-15]

Kode: 00001010 (0x0A), M[rd+immediate] = s1:[0-16]

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. source-1 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Transfer half word low 16 bits dari memori ke register (MOVML), M:[0-15]

Kode: 00001011 (0x0B), rd = M[s1+immediate]:[0-16]

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destinasi | Reg. source-1 | Immediate address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Transfer double word 64 bits register ke memori (MOVD)

Kode: 00001100 (0x0C)

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. destinasi | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Transfer double word 64 bits dari memori ke register (MOVMD)

Kode: 00001101 (0x0D)

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. destinasi | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Push/simpan register ke stack (PUSH)

Kode: 00001110 (0x0E)

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. destinasi | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Pop/keluarkan data dari stack, simpan ke register (POP)

Kode: 00001111 (0x0F)

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. destinasi | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

simpan flag register (FLAG) ke stack (PUSHF)

Kode: 00010000 (0x10)

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. destinasi | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Keluarkan data dari stack simpan ke flag register (POPF)

Kode: 00010001 (0x11)

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. destinasi | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Invoke Interrupt (INTR)

Kode: 00010010 (0x12)

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. destinasi | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Read input device (IN), perlu dikerjakan

Kode: 00010011 (0x13), IN rd, s1; operasi Input-output (simulasi berupa delay())

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. source-1 | Tidak digunakan |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Write output device (OUT) , perlu dikerjakan

Kode: 00010100 (0x14), OUT rd, s1; operasi Input-output (simulasi berupa delay())

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. source-1 | Tidak digunakan |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Unconditional Jump (JMP), untuk jump near

Kode: 00010101 (0x15), jmp addr; ke lokasi memori

|  |  |
| --- | --- |
| Op.code | Immediate/address |
| 8 bits | 24 bits (3 bytes) |
| Bit: 31-24 | 0-23 |
| Flag | Belum ditetapkan |

Unconditional Jump register (JMPR), untuk jump far

Kode: 00010110 (0x16), jmpr rd

|  |  |  |
| --- | --- | --- |
| Op.code | Reg. source-1 | Tidak digunakan |
| 8 bits | 4 bits | 20 bits |
| Bit: 31-24 | 20-23 | 0-19 |
| Flag | Belum ditetapkan | |

Conditional jump on equal (JE)

Kode: 00010111 (0x17), JE s1,s2,jump to immediate

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. source-2 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Conditional jump on not equal (JNE)

Kode: 00011000 (0x18), JNE s1,s2,jump to immediate

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. source-2 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Conditional jump on less than (JLT)

Kode: 00010111 (0x19), JLT s1,s2,jump to immediate, s1 < s2

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. source-2 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Conditional jump on greater than (JGT)

Kode: 00011000 (0x1A), JGT s1,s2,jump to immediate, s1 > s2

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. source-1 | Reg. source-2 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |

Arithmetic Subtract (SUB)

Kode: 00011011 (0x1B), rd = rs1 - rs2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Op.code | Reg destination | Reg. source-1 | Reg. source-2 |  |
| 8 bits | 4 bits | 4 bits | 4 bits | 12 bits |
| Bit: 31-24 | 20-23 | 16-19 | 15-12 | 0-12 |
| Flag | SF, ZF, CF, OF | | | |

Arithmetic subtract immediate (SUBI)

Kode: 00011100 (0x1C), rd = rs1 - immediate

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. source-1 | Immediate data |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | SF, ZF, CF, OF | | |

Transfer numeric data 16 bits dari memori ke register (MOVMHW)

Kode: 00011101 (0x1D), rd = M[s1 + immediate], data 16 bits

|  |  |  |  |
| --- | --- | --- | --- |
| Op.code | Reg. destination | Reg. Source-1 | Immediate/address |
| 8 bits | 4 bits | 4 bits | 16 bits |
| Bit: 31-24 | 20-23 | 16-19 | 0-15 |
| Flag | Belum selesai | | |