

# RISC

## (Reduced Instruction Set Computer)

Dr. Sutikno



**UNDIP** | UNIVERSITAS  
DIPONEGORO  
becomes an excellent research university

# RISC

- Jenis set intruksi, ada 2 jenis arsitektur komputer:
  - AK dengan Kumpulan perintah yang sederhana (*Reduced Instruction Set Computer* = RISC)
  - AK dengan Kumpulan perintah yang rumit (*Complex Instruction Set Computer* = CISC)
- RISC: penemuan penting dalam OAK, perubahan dramatis pada arsitektur prosesor.



# Pengertian RISC

- RISC: *Reduced Instruction Set Computer*
- RISC: pendekatan dalam arsitektur komputer yang bertujuan untuk menyederhanakan instruksi-instruksi dalam set instruksi komputer (instruction set)
- Fokus pada instruksi-instruksi yang paling dasar dan umum digunakan.



# Karakteristik Arsitektur RISC

- Siklus mesin ditentukan oleh **waktu yang digunakan** untuk mengambil **dua buah operand** dari register, melakukan **operasi ALU**, dan **menyimpan hasil operasinya** kedalam register
- Operasi berbentuk dari register-ke register yang hanya terdiri dari operasi load dan store yang mengakses memori.
- Penggunaan mode pengalamatan sederhana
- Penggunaan format-format instruksi sederhana, panjang instruksinya tetap dan disesuaikan dengan panjang word.



# Ciri-Ciri RISC

- Instruksi berukuran Tunggal. Umumnya berukuran 4 Byte
- Jumlah pengalamatan data sedikit ( $< 5$  buah)
- Tidak terdapat pengalamatan tidak langsung
- Tidak terdapat operasi yang menggabungkan operasi load/store dengan operasi aritmatika (terpisah)
  - Seperti penambahan ke memori dan penambahan dari memori



# Contoh processor menggunakan arsitektur RISC

- AMD29000, dikembangkan oleh Advanced Micro Devices (AMD).
- MIPS R2000, dikembangkan oleh MIPS Computer Systems (sekarang bagian dari Imagination Technologies) pada computer pribadi, server, system embeded
- SPARC, dikembangkan oleh Sun Microsystems (kemudian diakuisisi oleh Oracle).
- MC 88000, dikembangkan oleh Motorola.
- HP PA, dikembangkan oleh Hewlett-Packard (HP).
- IBM RT/TC, komputer workstation dan server
- IBM RS/6000, salah satu seri komputer workstation dan server
- intel i860, dikembangkan oleh Intel.
- Motorola 88000 (keluarga
- Motorola),
- PowerPC G5.



# Contoh intruksi 'MULT' pada RISC

## CISC

**MULT 2: 3, 5: 2**

## RISC

LOAD	A,	2:3
LOAD	B,	5:2
PROD	A,	B
STORE	2:3,	A

- LOAD A, 2:3: mengambil nilai dari alamat memori yang terletak di blok 2, offset 3, dan memuatnya ke dalam register A.
- LOAD B, 5:2
- PROD A, B: mengalikan isi register A dengan isi register B dan menyimpan hasilnya kembali ke dalam register A.
- STORE 2:3, A: menyimpan nilai dari register A ke alamat memori yang ditentukan



# CISC

- CISC: *Complex Intruction Set Computer*
- Processor yang memiliki set intruksi yang komplek dan lengkap.
- Processor memiliki set intruksi program yang lebih sedikit.





# Karakteristik CISC

- Sarat informasi memberikan keuntungan di mana ukuran **program-program yang dihasilkan akan menjadi relatif lebih kecil**, dan penggunaan memory akan semakin berkurang.
- Dimaksudkan untuk **meminimumkan jumlah perintah** yang diperlukan untuk mengerjakan pekerjaan yang diberikan. (Jumlah perintah sedikit tetapi rumit)  
Konsep CISC menjadikan mesin **mudah untuk diprogram dalam bahasa rakitan**



# Pendekatan CISC

- melaksanakan suatu perintah cukup dengan beberapa baris bahasa mesin sedikit mungkin.
- Contoh: **MULT 2:3, 5:2**
  - Perintah MULT : perintah untuk melakukan operasi perkalian antara dua nilai yang diambil dari dua alamat memori tertentu, kemudian menyimpan hasilnya kembali ke alamat memori yang lain.



# Karakteristik dari beberapa Prosesor CISC, RISC, dan Superskalar

- Prosesor superskalar: menjalankan beberapa instruksi secara parallel.

	Complex Instruction Set (CISC) Computer			Reduced Instruction Set (RISC) Computer		Superscalar		
Characteristic	IBM 370/168	VAX 11/780	Intel 80486	SPARC	MIPS R4000	PowerPC	Ultra SPARC	MIPS R10000
Year developed	1973	1978	1989	1987	1991	1993	1996	1996
Number of instructions	208	303	235	69	94	225		
Instruction size (bytes)	2-6	2-57	1-11	4	4	4	4	4
Addressing modes	4	22	11	1	1	2	1	1
Number of general-purpose registers	16	16	8	40 - 520	32	32	40 - 520	32
Control memory size (Kbits)	420	480	246	—	—	—	—	—
Cache size (KBytes)	64	64	8	32	128	16-32	32	64



# Operations

- Assignments
  - Movement of data
- Conditional statements (IF, LOOP)
  - Sequence control
- Procedure **call-return** is very time consuming
- Beberapa instruksi HLL menyebabkan banyak operasi kode mesin



# Relative Dynamic Frequency

	Dynamic Occurrence		Machine Instruction (Weighted)		Memory Reference (Weighted)	
	Pascal	C	Pascal	C	Pascal	C
Assign	45	38	13	13	14	15
Loop	5	3	42	32	33	26
Call	15	12	31	33	44	45
If	29	43	11	21	7	13
GoTo	-	3	-	-	-	-
Other	6	1	3	1	2	1



# Operands

- Terutama variabel skalar lokal
- Optimalisasi harus berkonsentrasi pada akses variabel lokal

	Pascal	C	Average
Integer constant	16	23	20
Scalar variable	58	53	55
Array/structure	26	24	25



# Eksekusi Intruksi

$$\text{Waktu eksekusi} = N \times S \times T$$

- N adalah jumlah perintah
- S adalah jumlah rata-rata langkah per perintah
- T adalah waktu yang diperlukan untuk melaksanakan satu Langkah
- Kecepatan eksekusi dapat ditingkatkan dengan menurunkan nilai dari ketiga variabel di atas.
- Arsitektur CISC berusaha menurunkan nilai **N**,
- Arsitektur RISC berusaha menurunkan nilai **S** dan **T**.
- Proses pipeline dapat digunakan untuk membuat nilai efektif S mendekati 1 (satu) artinya komputer menyelesaikan satu perintah dalam satu siklus waktu CPU.
- Nilai T dapat diturunkan dengan merancang perintah yang sederhana



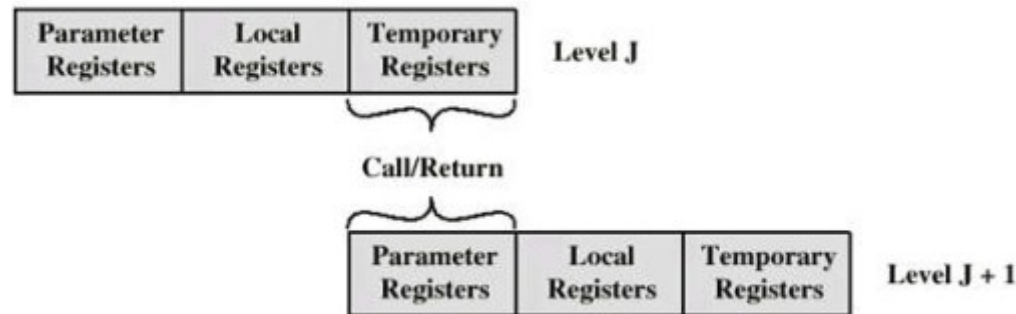
# Memperbesar file register

- Alasan: register merupakan perangkat penyimpanan yang paling cepat dibanding memori utama dan memori cache.
- Pendekatan dasar:
  - Pendekatan perangkat lunak yaitu compiler yang membutuhkan algoritka analisis program
  - Pendekatan perangkat keras yaitu memperbanyak jumlah register.





# Overlapping Register Windows



- Register dibagi menjadi beberapa window yang tumpang tindih.
- Setiap Window:
  - Parameter register: menyimpan parameter-parameter ketika fungsi dipanggil
  - Local register: menyimpan variable local saat fungsi/prosedur sedang dieksekusi.
  - Temporary register: menyimpan nilai sementara saat proses perhitungan dilakukan.
- Memungkinkan setiap fungsi memiliki kumpulan register sendiri untuk menyimpan parameter dan variable local tanpa menyimpan di memori utama.



# Variable global

- Untuk memenuhi kebutuhan penyimpanan variable global, yg diakses oleh lebih dari sebuah prosedur.
- Untuk memenuhi hal tersebut:
  - Variabel global pada HLL dapat disediakan lokasi-lokasi oleh compiler
    - Namun bagi yang sering mengakses, teknik tersebut tdk efisien.
  - Melibatkan Kumpulan register global didalam CPU.
    - Memiliki jumlah yang tetap
    - Dapat dipakai oleh semua prosedur



# Register vs Cache

Large Register File	Cache
<ul style="list-style-type: none"><li>• Semua skalar lokal</li></ul>	Skalar lokal yang baru dipakai
<ul style="list-style-type: none"><li>• Variabel2 individual</li></ul>	Sekelompok memori
<ul style="list-style-type: none"><li>• Variabel2 global yg di-assign kompiler</li></ul>	Variabel global yang baru dipakai
<ul style="list-style-type: none"><li>• Save/restore tergantung prosedur</li></ul>	Save/restore tergantung algoritma penggantian cache
<ul style="list-style-type: none"><li>• Pengalamatan Register</li></ul>	Pengalamatan Memori

- Perbedaan utama: **register** adalah unit penyimpanan internal yang sangat cepat di dalam prosesor, sementara **cache** adalah memori eksternal yang berada di antara prosesor dan memori utama, bertindak sebagai penyimpanan sementara untuk data yang sering digunakan.
- Sistem berbasis cache akan lebih lambat.



# Pipelining

- Teknik yang digunakan untuk meningkatkan kinerja processor dengan mengizinkan beberapa intruksi untuk dieksekusi secara bersamaan.
- Proses eksekusi intruksi dibagi menjadi beberapa tahapan, dan setiap tahapan dilakukan secara bersamaan.



# Pipelining RISC

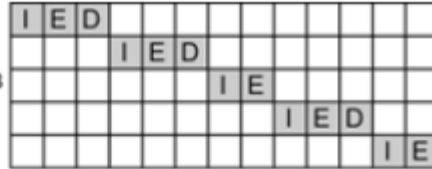
## (Instruksi register to register)

- Siklus instruksi memiliki 2 fase:
  1. I: Instruction fetch (Pengambilan instruksi)
  2. E: Execute (melakukan operasi ALU dengan register input dan output)
- Operasi Load dan Store memiliki 3 fase:
  1. I
  2. E
  3. D: Memori (operasi register ke memori atau memori ke register)



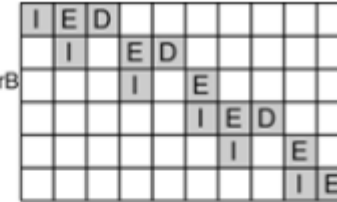
# Efek Pipelining

Load  $rA \leftarrow M$   
 Load  $rB \leftarrow M$   
 Add  $rC \leftarrow rA + rB$   
 Store  $M \leftarrow rC$   
 Branch X



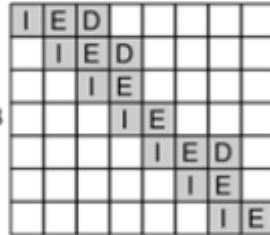
(a) Sequential execution

Load  $rA \leftarrow M$   
 Load  $rB \leftarrow M$   
 Add  $rC \leftarrow rA + rB$   
 Store  $M \leftarrow rC$   
 Branch X  
 NOOP



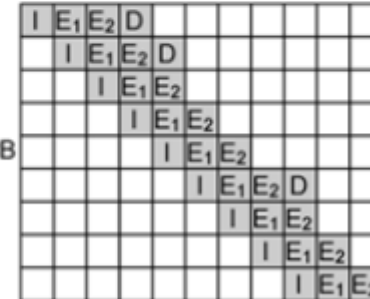
(b) Two-stage pipelined timing

Load  $rA \leftarrow M$   
 Load  $rB \leftarrow M$   
 NOOP  
 Add  $rC \leftarrow rA + rB$   
 Store  $M \leftarrow rC$   
 Branch X  
 NOOP



(c) Three-stage pipelined timing

Load  $rA \leftarrow M$   
 Load  $rB \leftarrow M$   
 NOOP  
 NOOP  
 Add  $rC \leftarrow rA + rB$   
 Store  $M \leftarrow rC$   
 Branch X  
 NOOP  
 NOOP



(d) Four-stage pipelined timing

- Branch: pencabangan
- NOOP: No operation, untuk mengakomodasi delay data.
- E1: membaca file register
- E2: menulis operasi ALU



# Kelebihan Rancangan RISC

- Kinerja Sistem CPU
- Optimalisasi dan pegefektifan computer
- Dengan menggunakan instruksi yang sederhana terdapat kemungkinan untuk:
  - Memindahkan fungsi –fungsi keluar loop
  - Melakukan reorganisasi kode untuk efisiensi
  - Memaksimalkan pemakaian register
  - Melakukan perhitungan bagian instruksi kompleks pada saat kompilasi
  - Memudahkan kerja unit kontrol
  - Memudahkan implementasi pipelining



# Kelebihan Rancangan RISC

- Implementasi Perangkat Keras
  - Kesederhanaan instruksi dan unit control menghasilkan hardware sederhana
  - Hardware sederhana dapat diletakkan dalam satu keping tunggal
  - Hardware sederhana menghasilkan proses hardware dengan dimensi yang lebih kecil, konsumsi daya rendah dan lebih ekonomis.



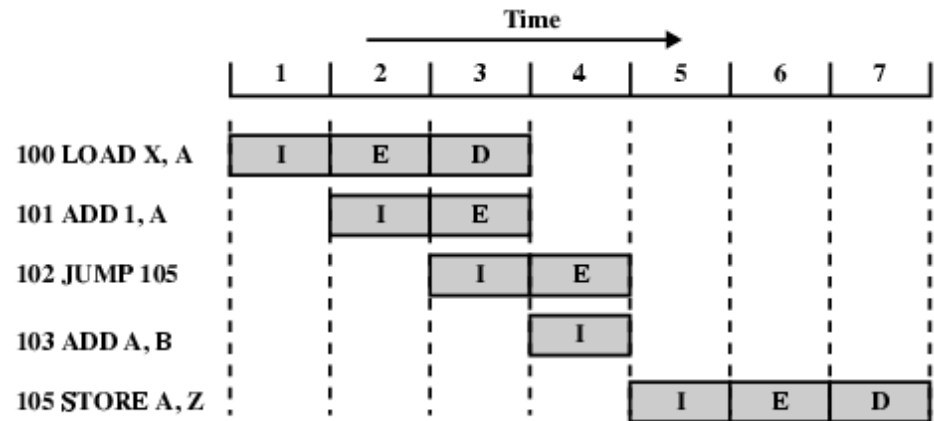


# Normal and Delayed Branch

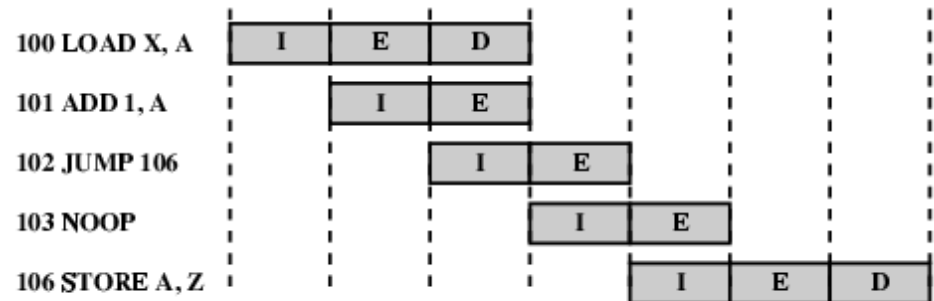
Address	Normal	Delayed	Optimized
100	LOAD X,A	LOAD X,A	LOAD X,A
101	ADD 1,A	ADD 1,A	JUMP 105
102	JUMP 105	JUMP 105	ADD 1,A
103	ADD A,B	NOOP	ADD A,B
104	SUB C,B	ADD A,B	SUB C,B
105	STORE A,Z	SUB C,B	STORE A,Z
106		STORE A,Z	



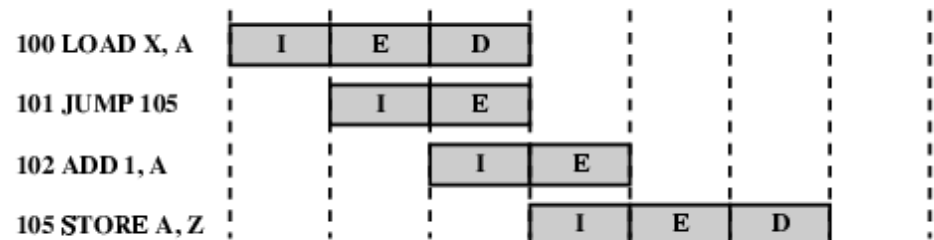
# Use of Delayed Branch



(a) Traditional Pipeline



(b) RISC Pipeline with Inserted NOOP



(c) Reversed Instructions



# Soal

1. Jelaskan apa yang dimaksud dengan arsitektur RISC dan sebutkan minimal 3 karakteristik utama dari arsitektur ini?
2. Bandingkan arsitektur RISC dengan CISC dalam hal pendekatan eksekusi instruksi dan penggunaan memori. Berikan contoh instruksi yang menunjukkan perbedaan tersebut?
3. a. Apa yang dimaksud dengan pipelining pada arsitektur RISC?  
b. Jelaskan bagaimana siklus instruksi bekerja pada instruksi 'load' dan 'store' dalam arsitektur RISC dengan pipelining?
4. Mengapa register dianggap lebih cepat dibandingkan cache dan memori utama? Jelaskan pula dua pendekatan yang digunakan untuk memperbesar file register?
5. Sebutkan dan jelaskan minimal tiga keuntungan dari penerapan rancangan RISC terhadap sistem CPU dan implementasi perangkat keras?



# TERIMAKASIH



**UNDIP** | UNIVERSITAS  
DIPONEGORO  
becomes an excellent research university