



PROGRAM STUDI

**S1 SISTEM KOMPUTER**

**UNIVERSITAS DIPONEGORO**

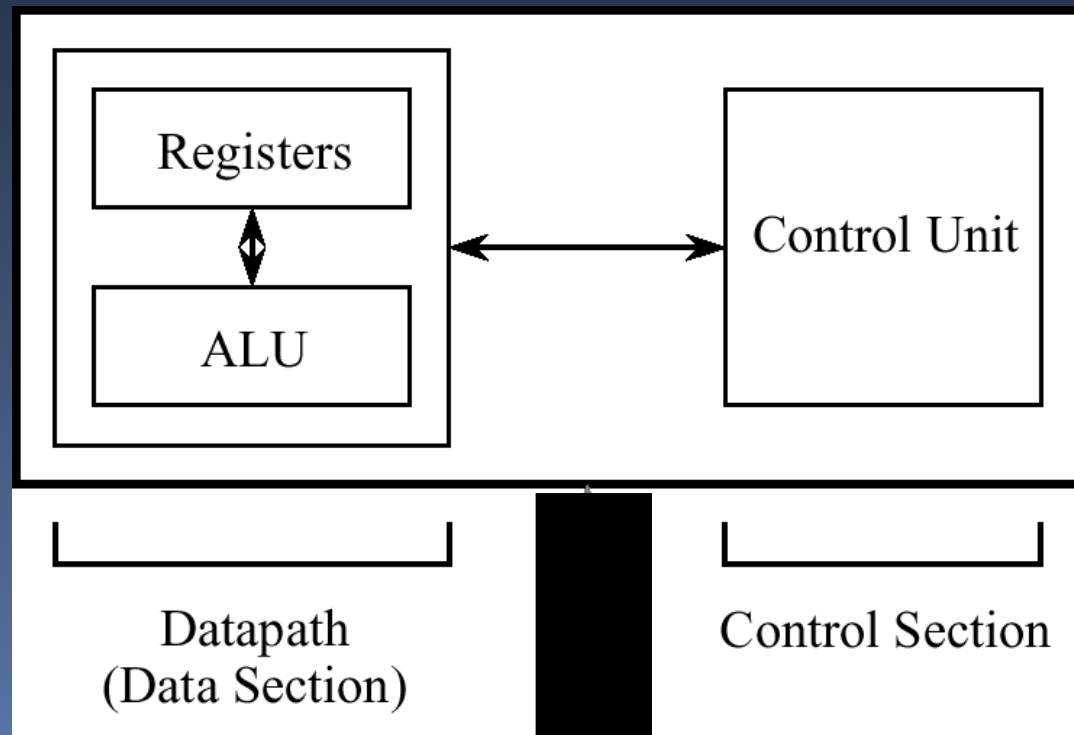
# ***CENTRAL PROCESSING UNIT*** **CPU**

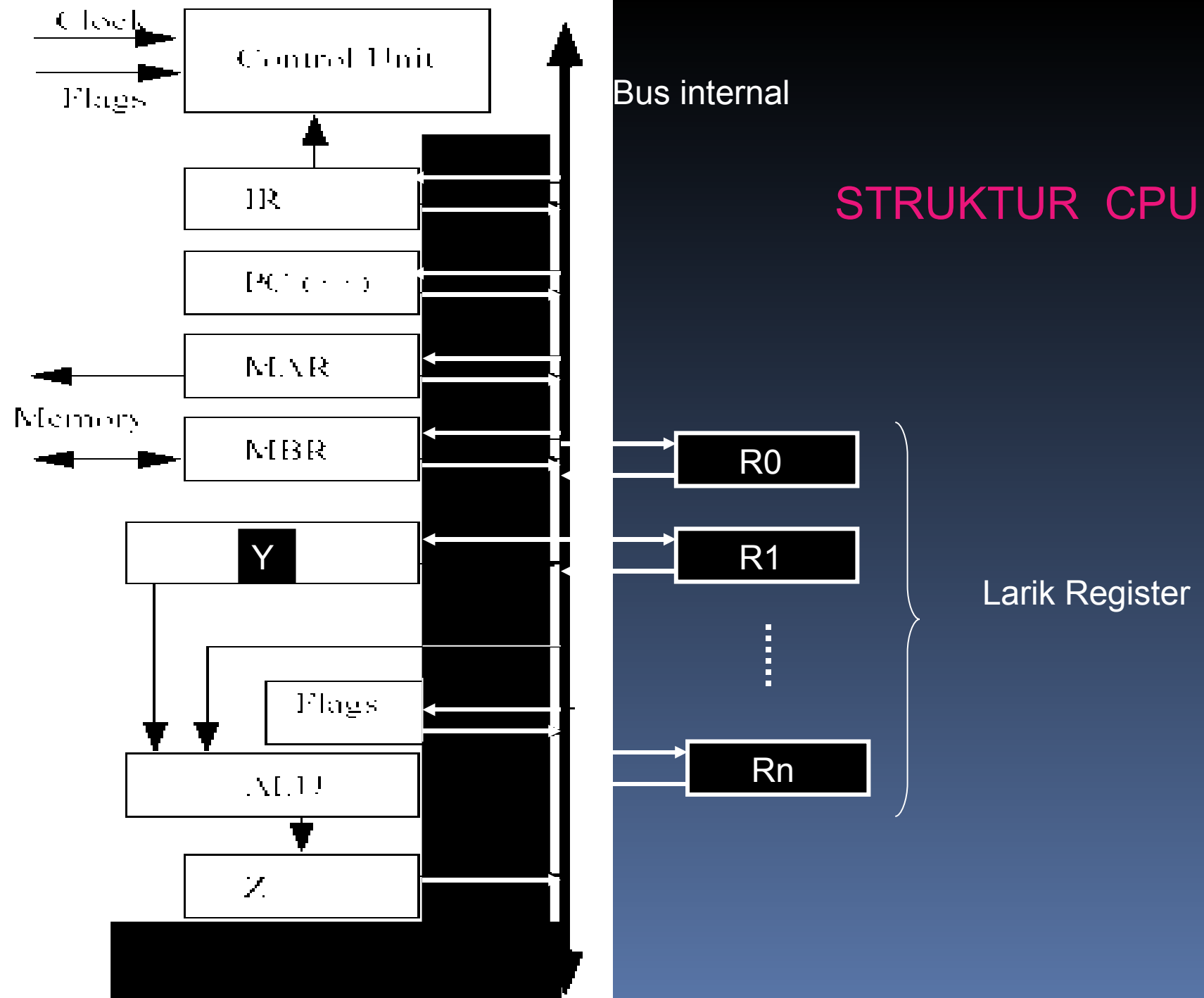
Oky Dwi Nurhayati, ST, MT  
email: [okyd@undip.ac.id](mailto:okyd@undip.ac.id)

# Central Processing Unit

CPU terdiri dari :

- Bagian data (Datapath) yang berisi register register untuk penyimpanan data sementara dan sebuah ALU untuk melaksanakan operasi operasi aritmatika dan logika
- Bagian kendali yang bertugas menginterpretasikan instruksi dan membangkitkan sinyal sinyal kendali agar operasi dalam instruksi tsb dilaksanakan.





# Register register dalam CPU

Kelompok kelompok register :

- Register register yang *user-visible* :  
Dapat diakses (dibaca dan diubah) oleh pemrogram melalui instruksi instruksi mesin.
4. Register register Kontrol dan Status

Register register yang *user-visible* :

- **General Purpose Register**  
Dapat digunakan untuk menyimpan data ataupun untuk menyimpan atau untuk kalkulasi alamat operand.
- **Register Data**  
Hanya dapat digunakan untuk menyimpan data
- **Register Alamat**  
Hanya dapat digunakan untuk menyimpan alamat atau untuk kalkulasi alamat operand.  
Contoh : Base Registers, Segment Register, Index Register, Stack Pointer

# Register register Kontrol dan Status

- Register register untuk komunikasi dengan unit unit diluar CPU

MAR (Memory Address Register) untuk menyatakan alamat lokasi

operand dalam memory yang akan dibaca atau ditulisi oleh CPU.

MBR atau MDR (Memory Buffer atau Data Register) merupakan tempat penyimpanan (sementara) data yang baru saja dibaca dari atau data yang akan dituliskan ke memory

PC (Program Counter) untuk menyatakan alamat lokasi instruksi yang akan dibaca oleh CPU dari memory.

- Register instruksi :

IR (Instruction Register) digunakan untuk menyimpan instruksi yang sedang dieksekusi dalam CPU.



Instruction Buffer untuk menyimpan instruksi instruksi yang menunggu giliran untuk dieksekusi dalam CPU.

- Register Status (PSW = Processor Status Word, atau Flag Register)

Terdiri dari bit bit (flag) yang menyatakan status CPU.



**Interrupt Enable/Disable** (Interrupt flag)

I = 1 CPU tidak akan melayani interupsi

I = 0 CPU akan melayani interupsi

**Sign Flag**

S = 0 hasil operasi ALU positif

S = 1 hasil operasi ALU negatif

**Zero Flag**

Z = 1 hasil operasi ALU = 0

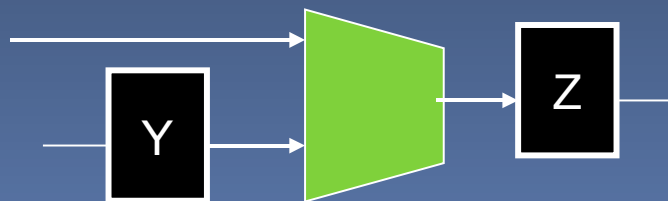
Z = 0 hasil operasi ALU  $\neq$  0

**Supervisor Flag** : CPU dalam mode sistem atau mode user

- Register register ALU

Register Y sebagai buffer bagi operand source pada operasi ALU.

Register Z sebagai buffer bagi data hasil operasi ALU, disebut akumulator



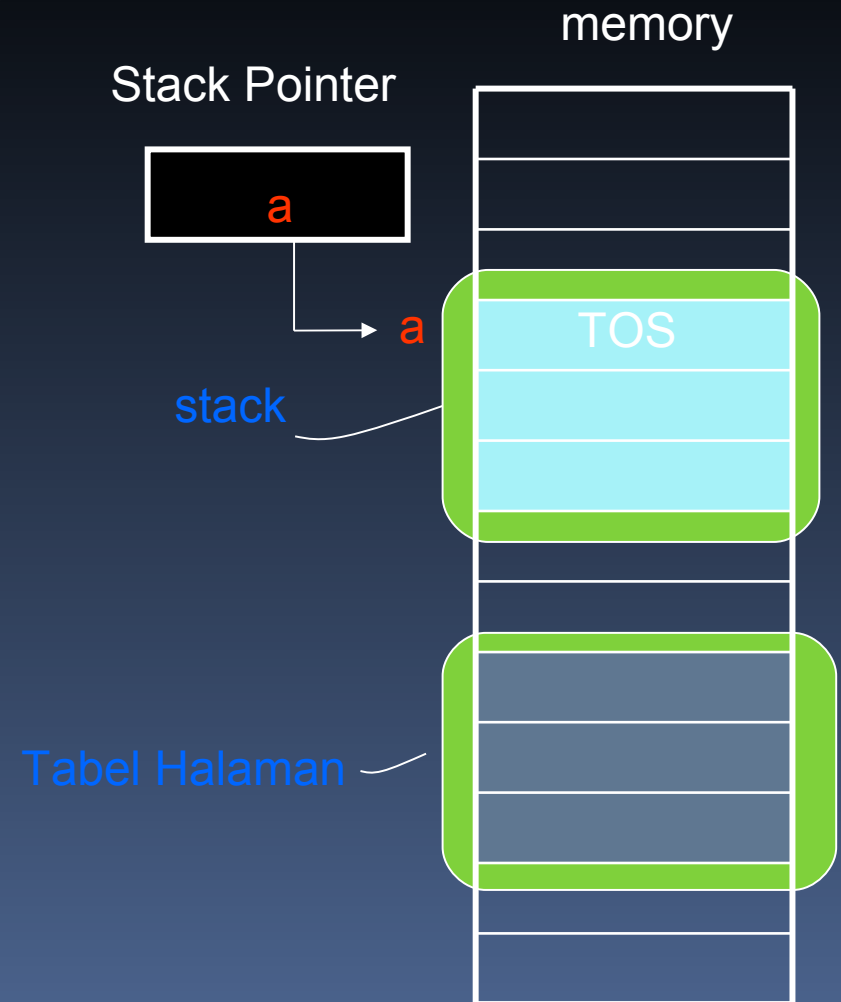
## Stack Pointer

- Stack adalah tempat penyimpanan data yang aksesnya diorganisasikan secara LIFO (Last In First Out).
- Stack dapat diimplementasikan sebagai satu set lokasi didalam main memory atau dengan menggunakan 1 set register didalam CPU
- Stack Pointer menunjukkan alamat bagian teratas stack (Top of Stack)
- Ada dua jenis stack : user stack dan system stack

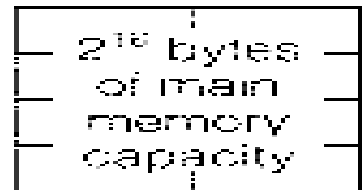
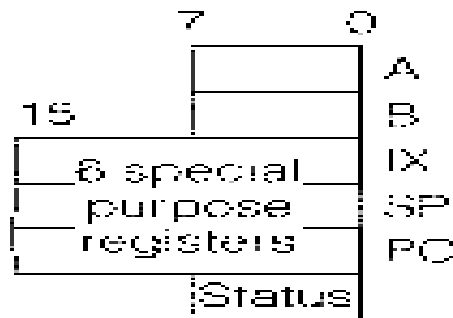
## Page Table Pointer

- Menunjukkan alamat Tabel Halaman (utk pengelolaan sistem pengingat)

## Register register pada Control Unit

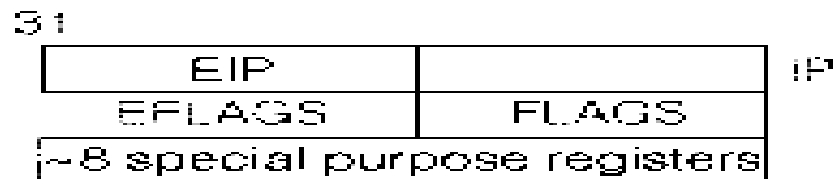
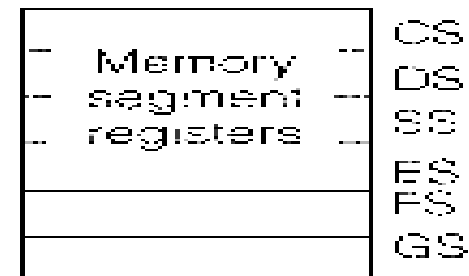
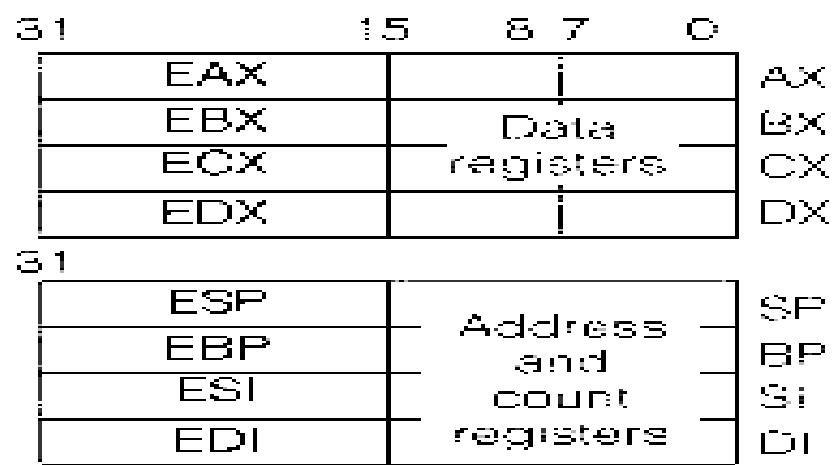


**M6800**  
(1975)



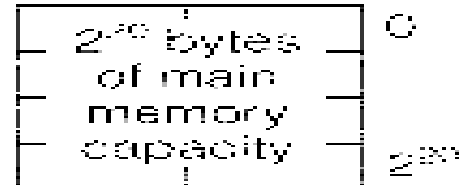
Fewer than 100 instructions

**Pentium 4**  
(2000)



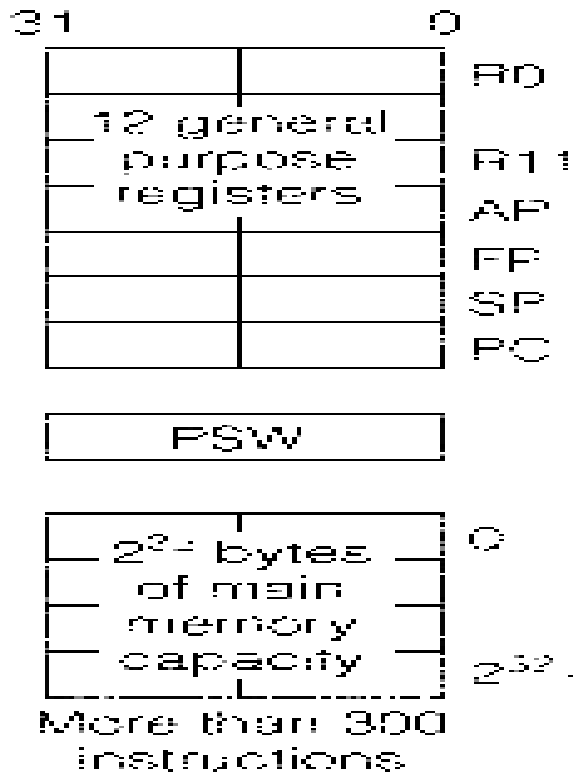
8 80-bit floating point reg's.  
8 64-bit MMX registers  
8 128-bit vector registers

More than 120 instructions (200 for Pentium)

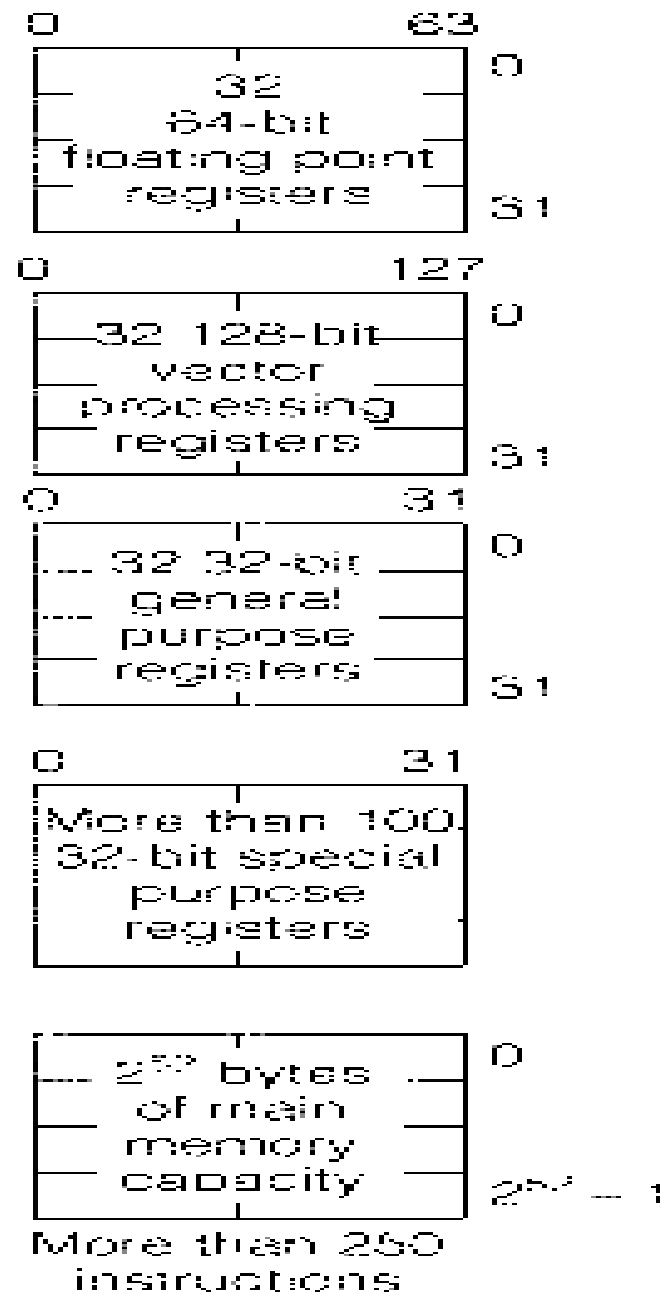




**VAX11**  
(1981)

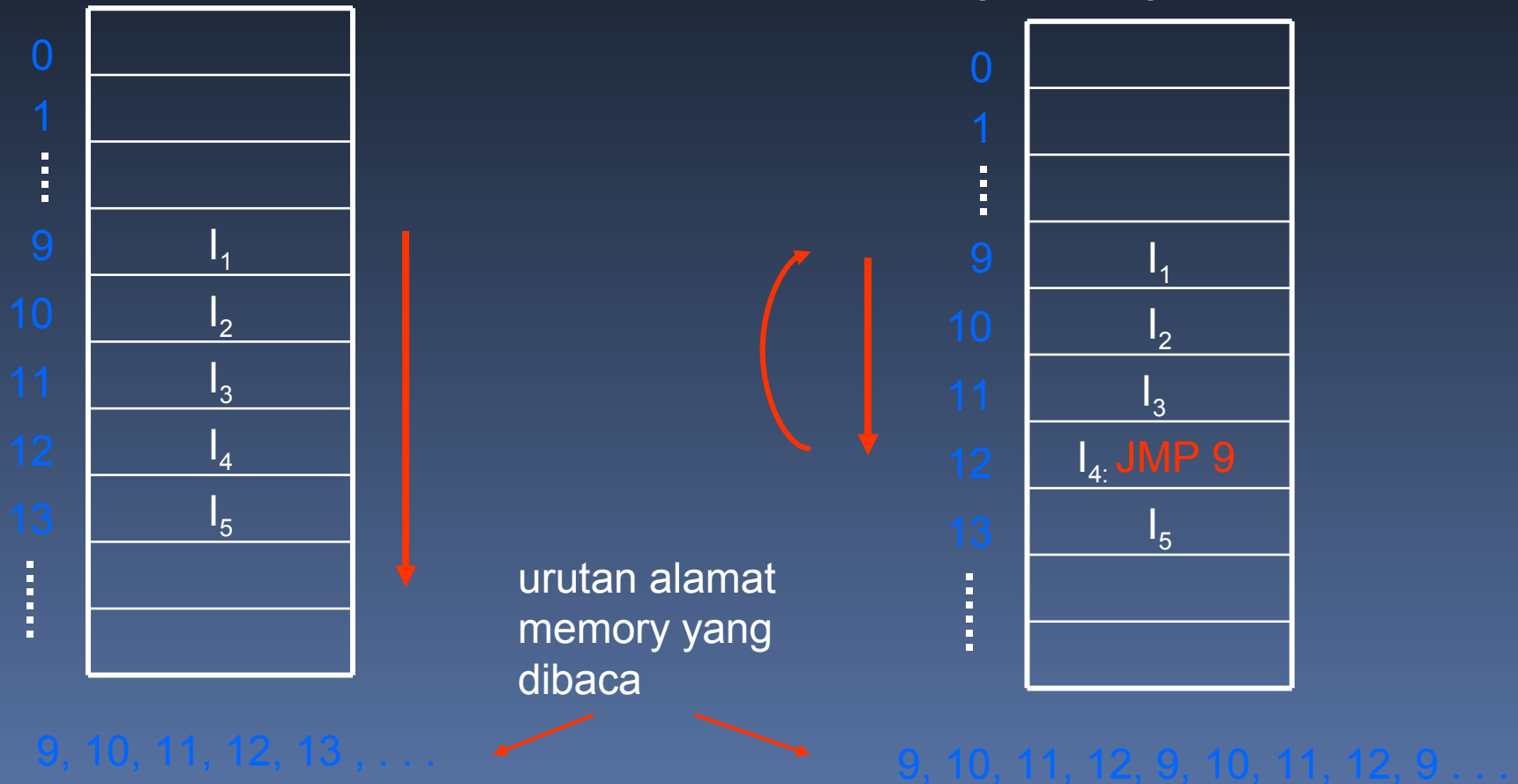


**PPC G4**  
(1999)

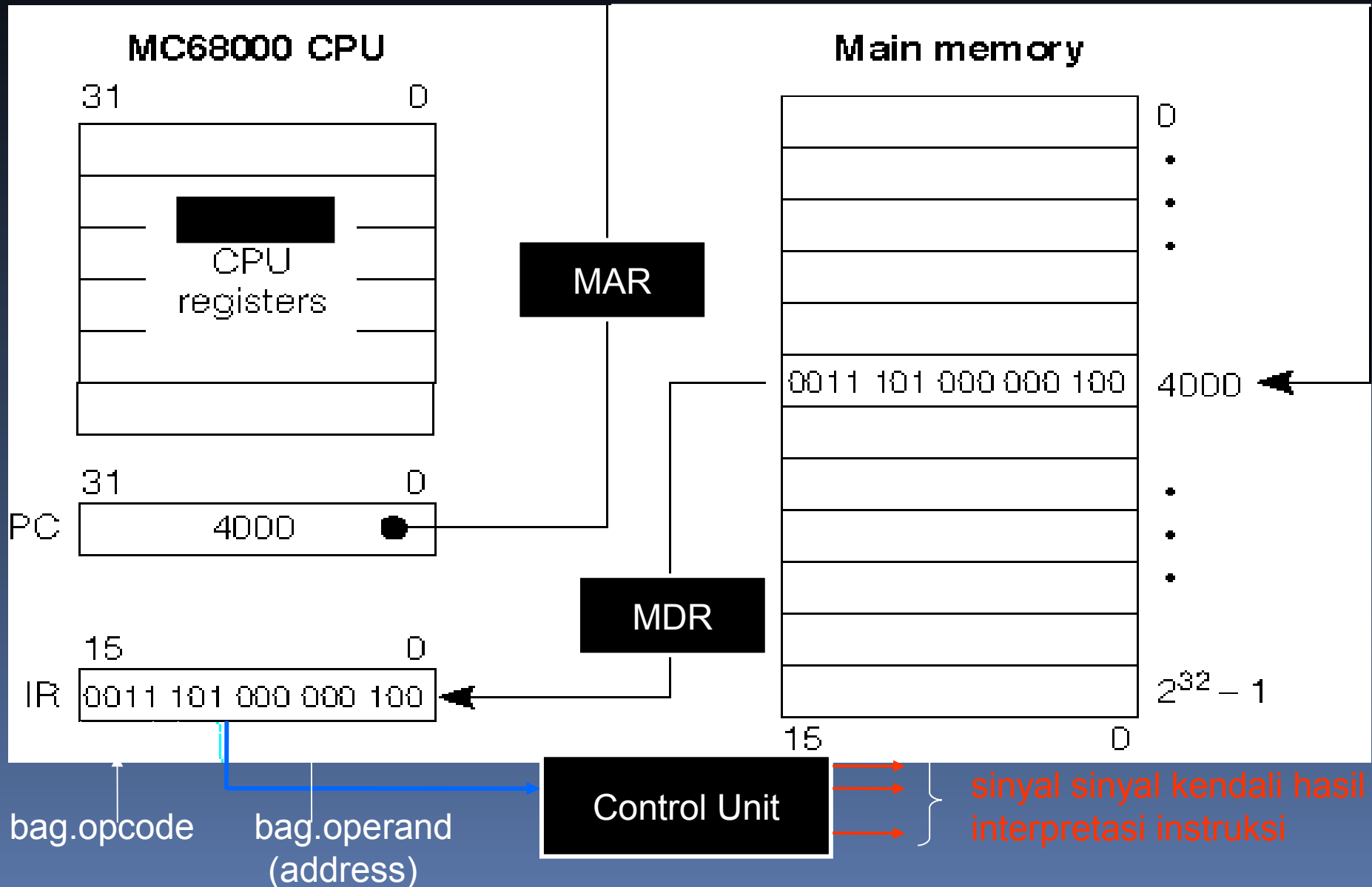


# EKSEKUSI PROGRAM

- Sebuah program tersusun dari instruksi instruksi.
- Eksekusi sebuah program = eksekusi instruksi instruksi penyusun program tersebut.
- Setiap instruksi akan dieksekusi oleh CPU dalam satu instruction cycle
- Urutan eksekusi instruksi ditentukan oleh logika program



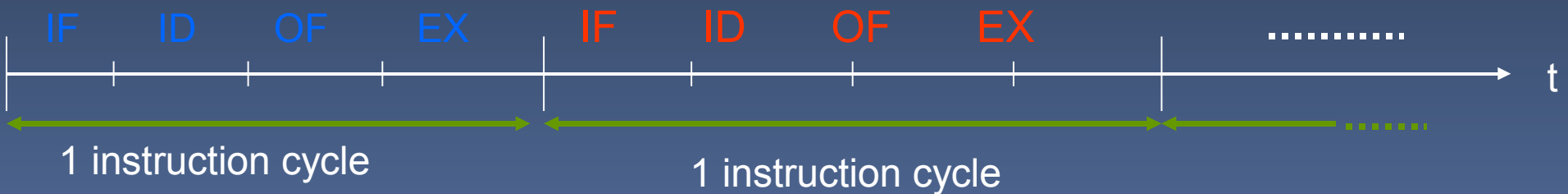
Alamat lokasi penyimpanan instruksi yang akan dibaca dari memory ditunjukkan oleh isi Program Counter (PC).



# Instruction Cycle (Fetch-Execute Cycle)

Eksekusi 1 instruksi dilakukan dalam 1 instruction cycle, yang terdiri dari langkah langkah sbb :

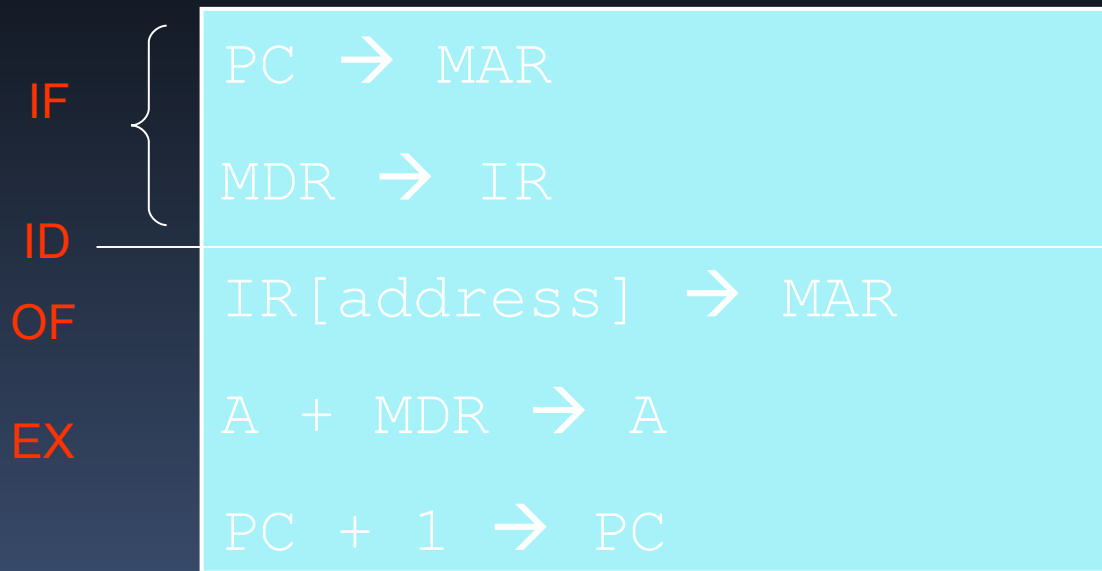
- (1) Baca (*Fetch*) instruksi yang akan di eksekusi dari memory ke CPU.
- (2) Interpretasikan (*Decode*) opcode dari instruksi tersebut.
- (3) Baca operand dari memory (bila instruksi tsb melibatkan operand yang ada di memory)
- (4) Laksanakan (*Execute*) instruksi dan simpan hasilnya (kalau ada).
- (5) Ulangi langkah 1 untuk instruksi berikutnya.



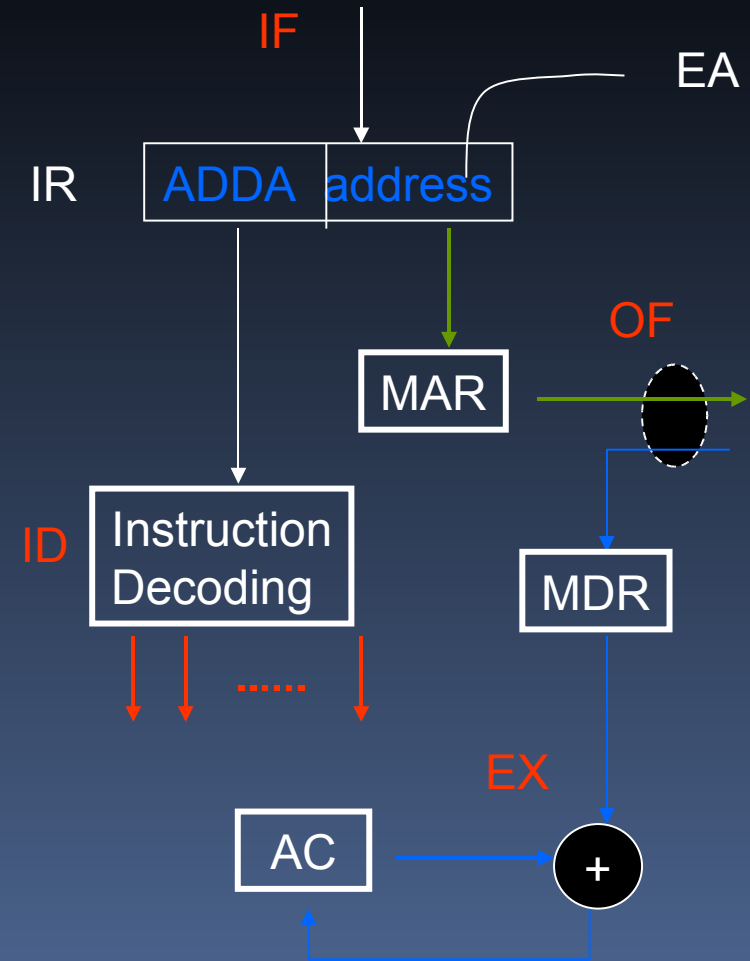
IF : *Instruction Fetch*  
OF : *Operand Fetch*

ID : *Instruction Decoding*  
EX : *Execute*

Eksekusi instruksi penjumlahan operand dari memory dengan isi akumulator, mode pengalamatan langsung.

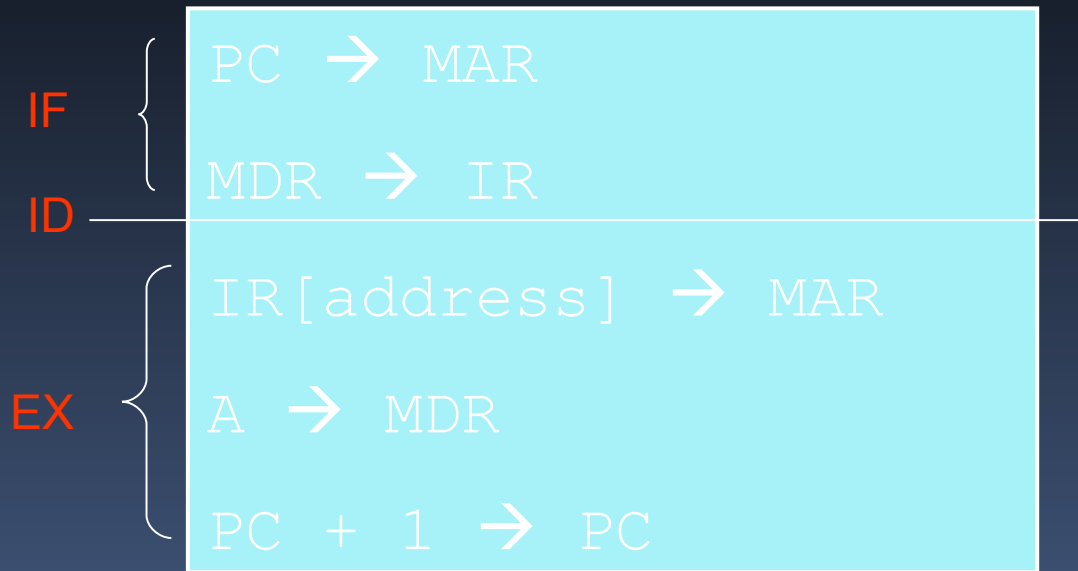


alamat instruksi berikutnya

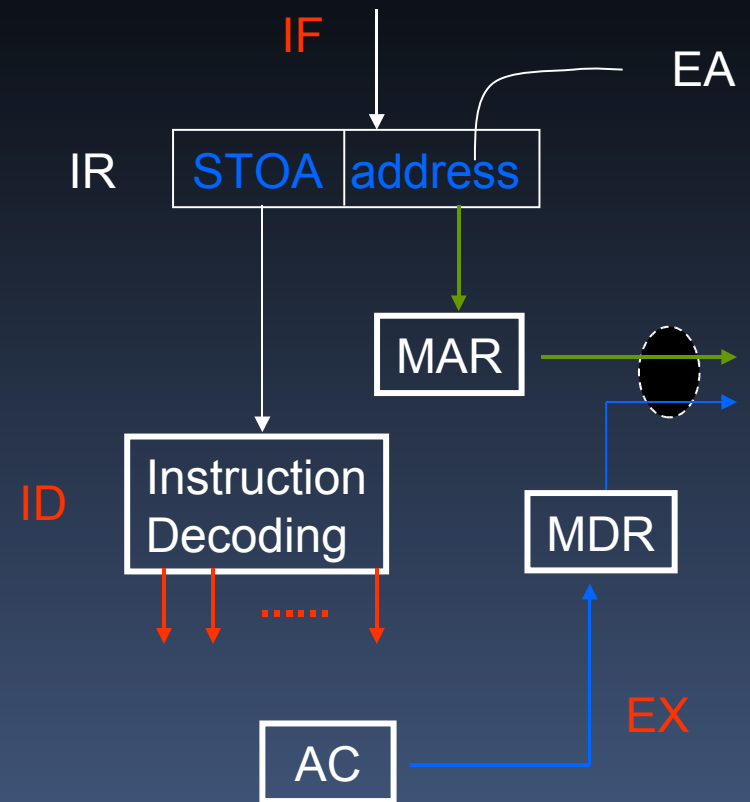


Untuk mode pengalamatan tak langsung ?

Eksekusi instruksi untuk menyimpan isi akumulator ke memory dengan mode pengalamatan langsung.



tidak ada OF



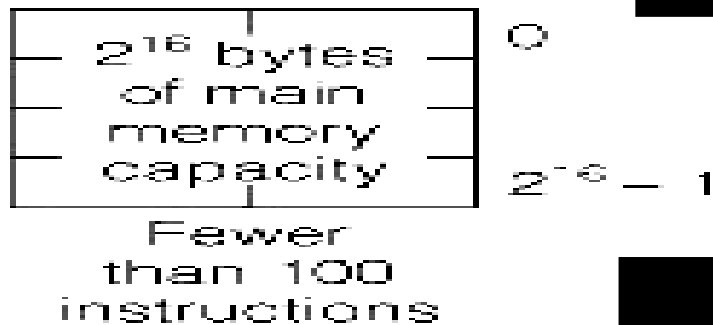
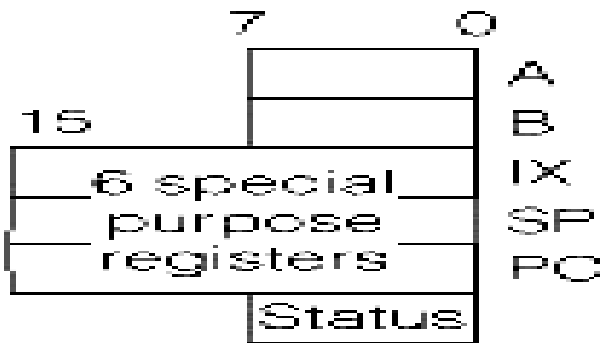
# Program sederhana, penjumlahan (A + B)

Location	Instruction		Comment
	Hex code	MNEMONIC	
000	0004	LOAD (004)	Copy A to accumulator
001	2005	ADD (005)	Add B to accumulator
002	1006	STORE (006)	Store (A + B) in location 006
003	F000	STOP	Stop
004	.....		A
005	.....		B
006	.....		SUM of A and B

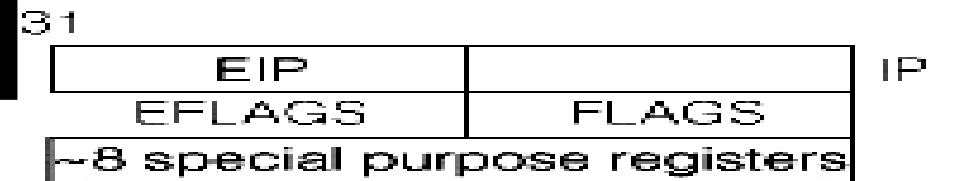
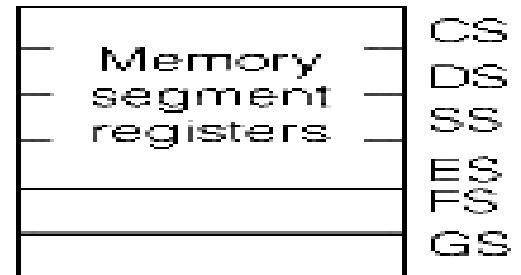
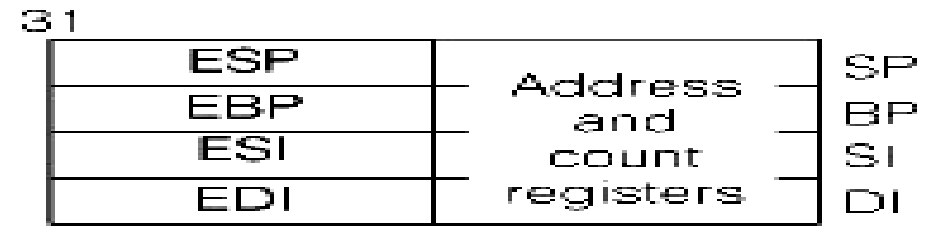
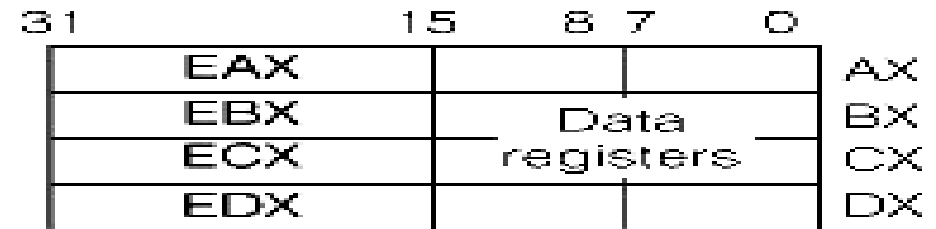
# Programmer's Models

## 4 commercial machines

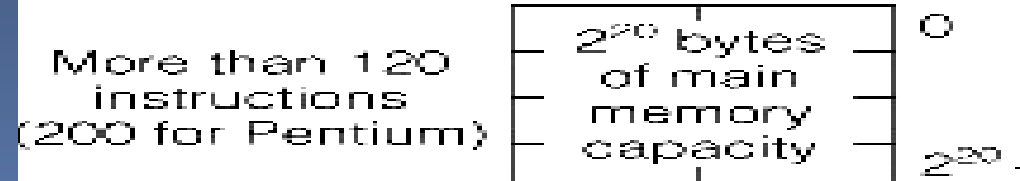
### M6800 (1975)



### Pentium 4 (2000)

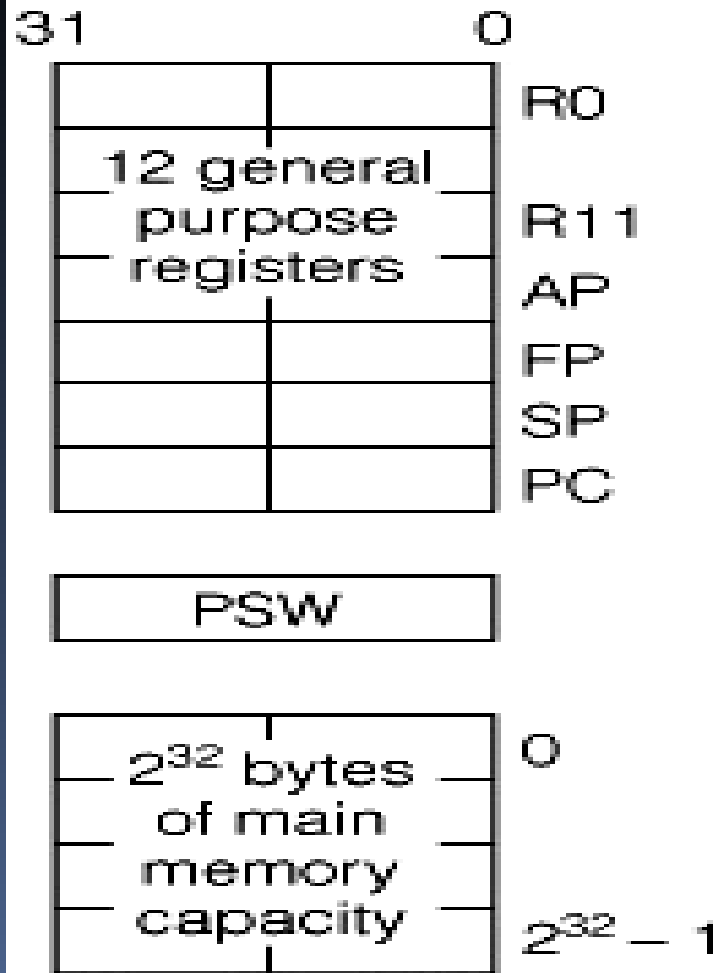


8 80-bit floating point reg's.  
8 64-Bit MMX registers  
8 128-bit vector registers





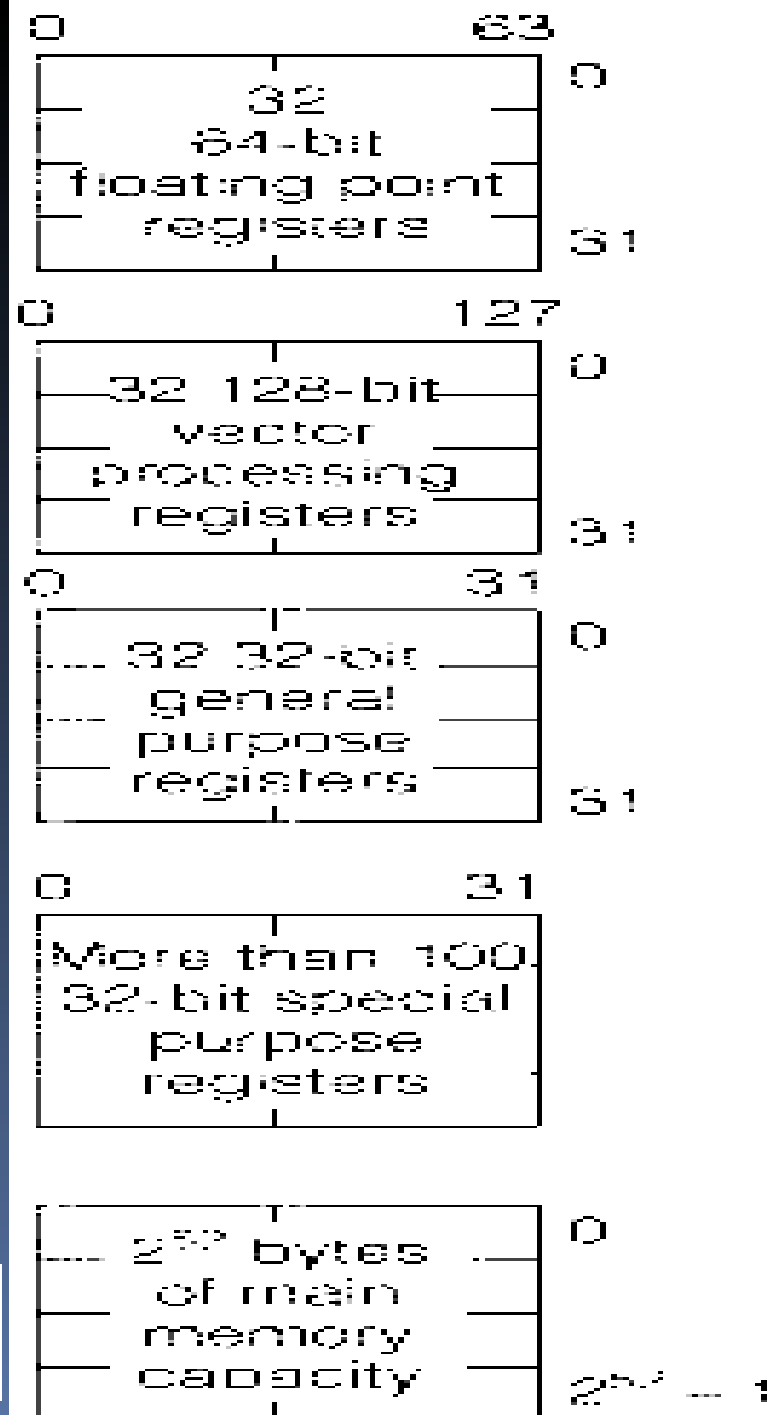
# VAX11 (1981)



More than 300 instructions

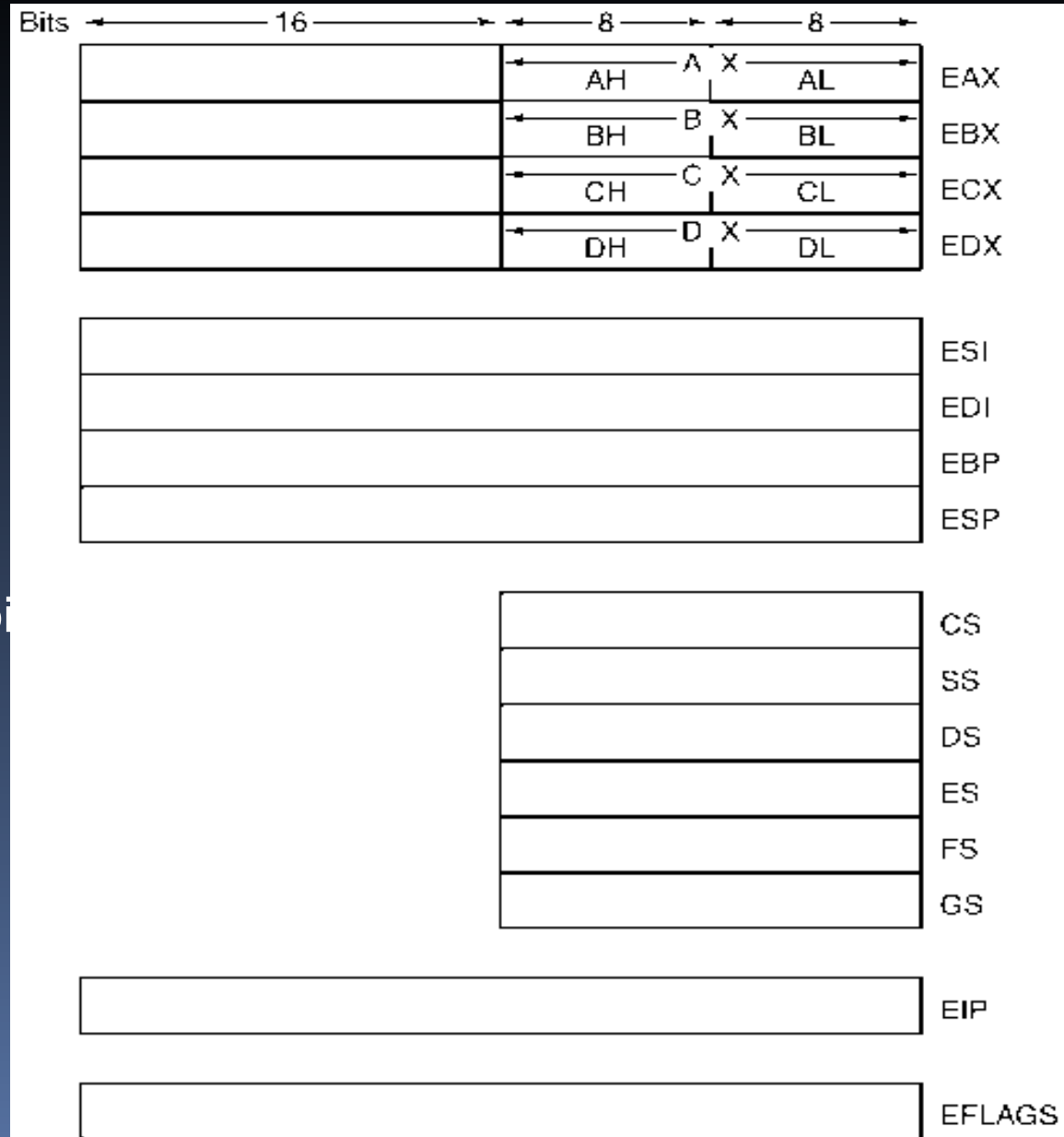
# PPC G4 (1999)

More than 250 instructions



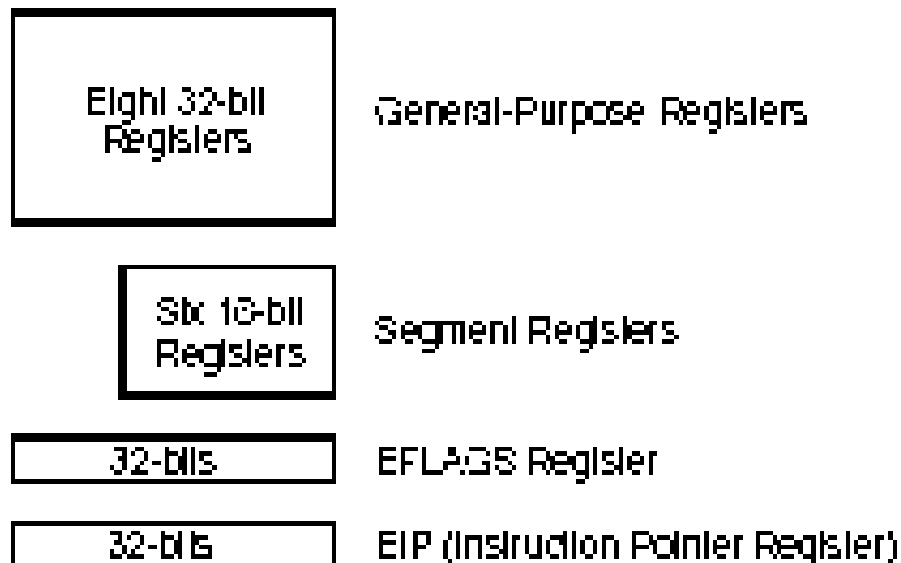
More than 250 instructions

# Registers 80x86

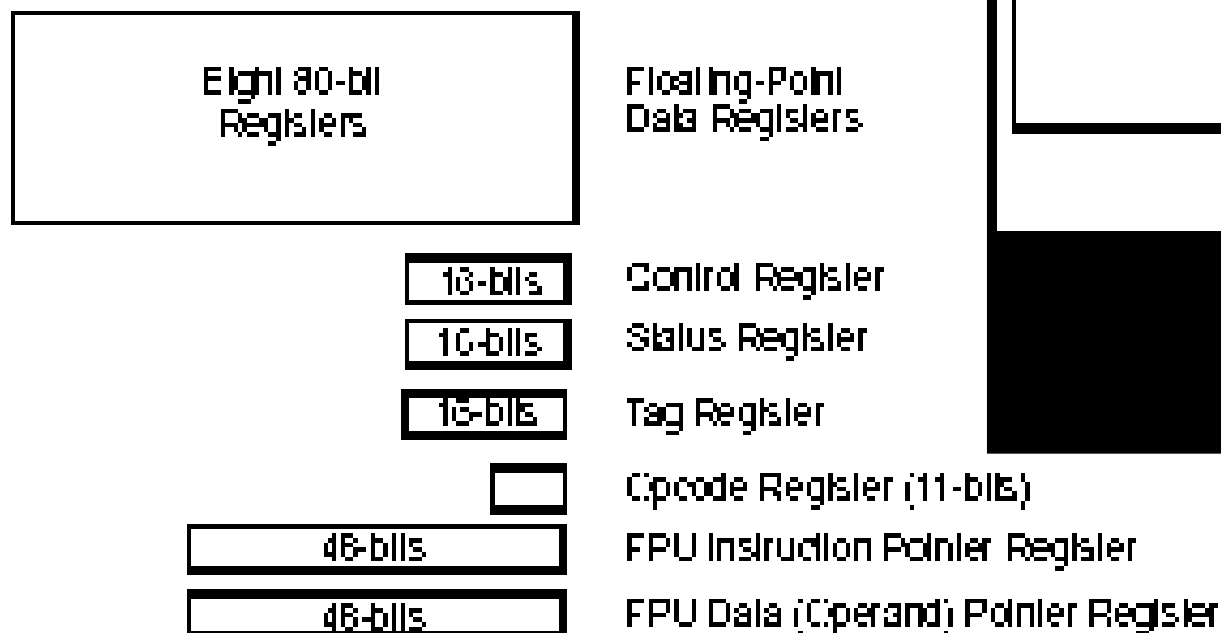


- PSW (Eflags)
  - N - result negative
  - Z - result Zero
  - V - overflow
  - C - carry out of high order bit
  - A - carry out of bit 3
  - P - even parity

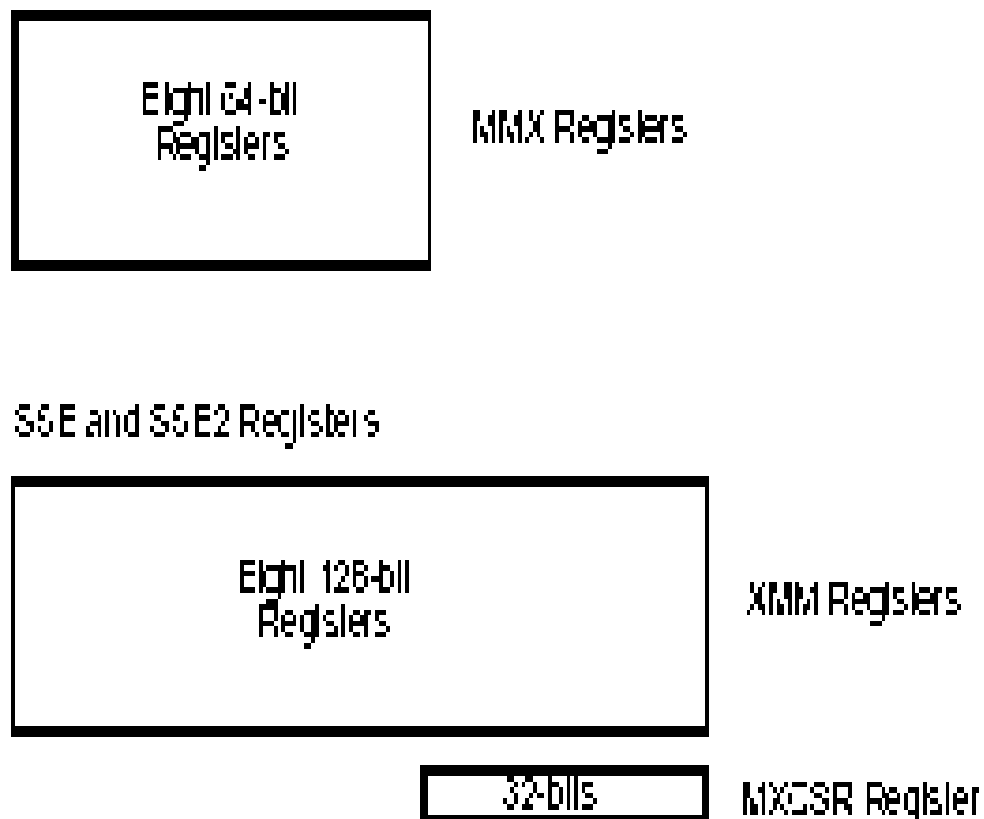
## Basic Program Execution Registers



## FPU Registers



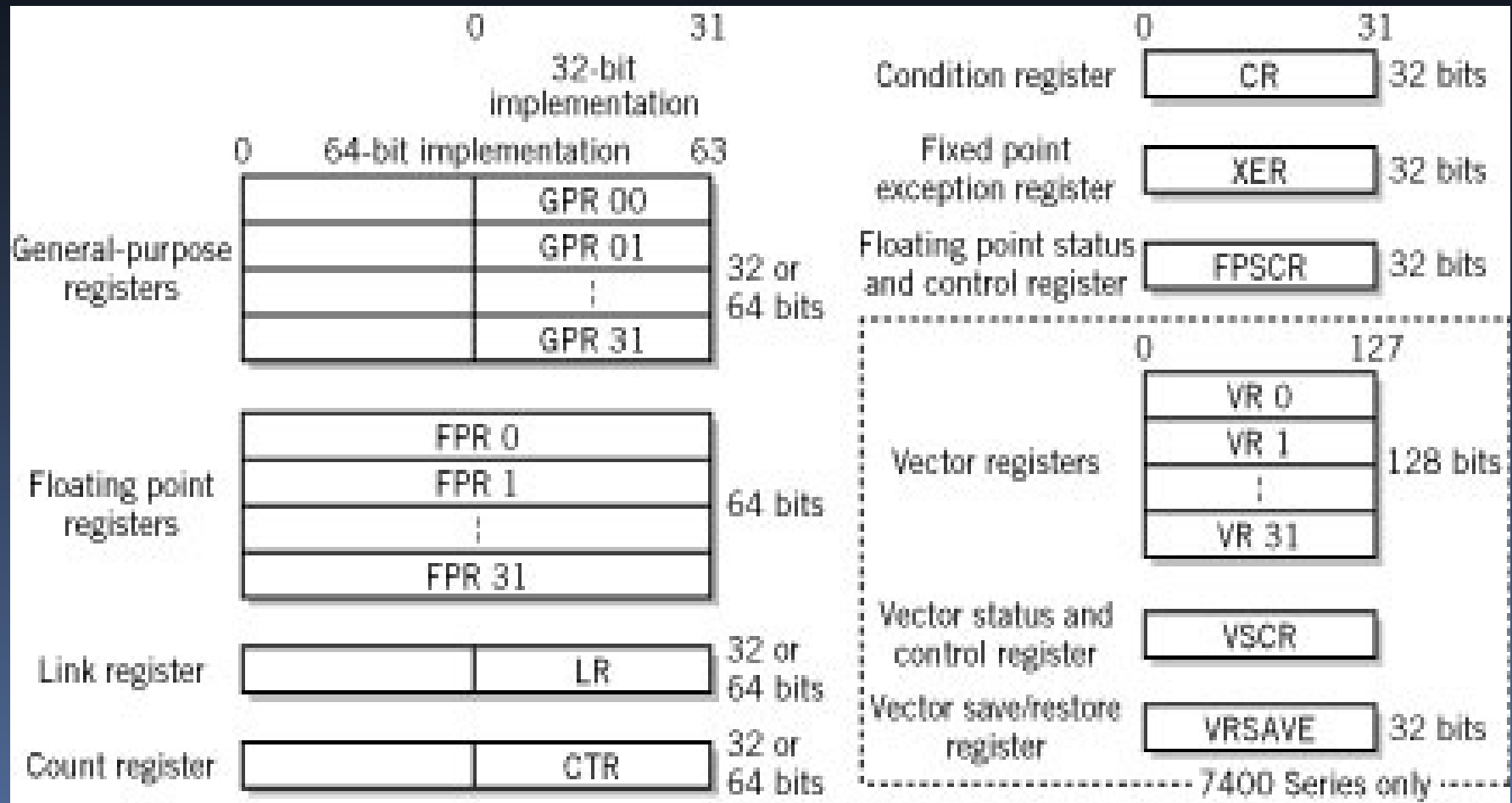
## MMX Registers



PENTIUM IV  
Lingkungan Eksekusi

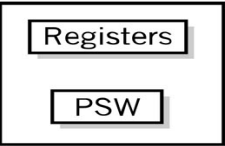


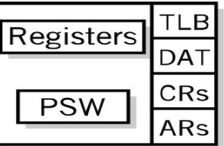


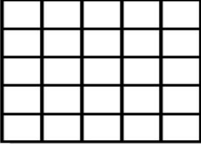
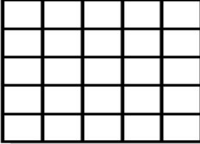
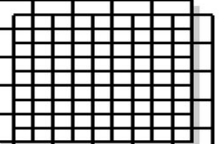
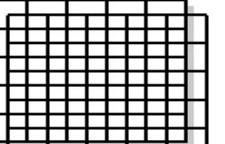
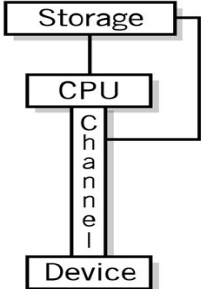
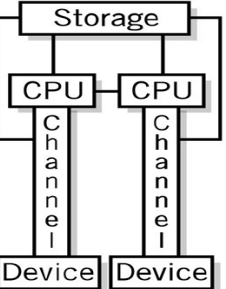
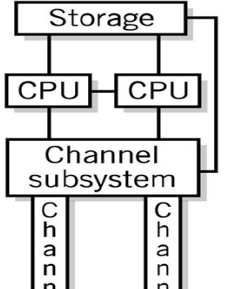
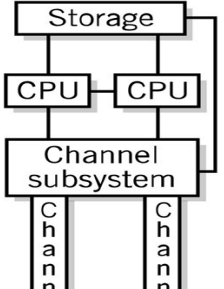
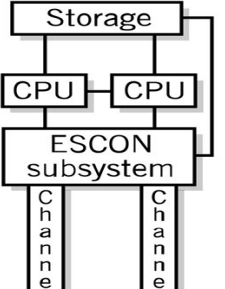
Dasar

# Power PC



IBM

# Architectural Evolution of 360/370/390/zSeries Computers

	360	370	370/XA	ESA/370	390 and zSeries
CPU	 <p>CPU</p>	 <p>CPU 1, 2</p>	 <p>CPU 1..n</p>	 <p>CPU 1..n</p>	 <p>CPU 1..n</p>
Storage	 <p>Real</p>	 <p>Virtual</p>	 <p>Virtual</p>	 <p>Virtual with multiple address space</p>	 <p>Virtual with multiple address space</p>
System block diagram					
Major developments		<ul style="list-style-type: none"> <li>• Virtual storage</li> <li>• Dual CPU processing</li> <li>• Control registers</li> </ul>	<ul style="list-style-type: none"> <li>• New I/O architecture</li> <li>• 31-bit addressing</li> <li>• Improved multi-processing</li> </ul>	<ul style="list-style-type: none"> <li>• Multiple address spaces</li> <li>• Access registers</li> <li>• Expanded storage</li> </ul>	<p><b>390</b></p> <ul style="list-style-type: none"> <li>• ESCON, FICON</li> <li>• Sysplex facility</li> <li>• Logical partitioning</li> </ul> <p><b>zSeries</b></p> <ul style="list-style-type: none"> <li>• 64-bit processing, registers, addresses</li> <li>• Gigabit ethernet</li> <li>• HyperSockets</li> <li>• Dynamic virtual IP addressing</li> </ul>

