

# Performance

Team Dosen  
Telkom University  
2016

# Definisi Performa

Pesawat	Kapasitas (orang)	Jarak Tempuh (mil)	Kecepatan (mil/jam)	Berat (kg)
Boeing 777	375	4630	610	228.750
Boeing 747	470	4150	610	268.700
BAC/Sud Concorde	132	4000	1350	178.200
Douglas DC-8-50	146	8720	544	79.424

# Perkembangan Komputer

Performansi ↑

Kapasitas ↑

Ukuran ↓

Harga ↓

# Generasi Komputer

Tabung Hampa  
(1946 – 1957)

Transistor  
(1958 – 1964)

Small &  
Medium scale  
integration  
(1965-1971)

- sampai 3,000 komponen pada satu chip

Large scale  
integration  
(1971-1977)

- 3,000 - 100,000 komponen pada satu chip

Very large scale  
integration  
(1978 – now)

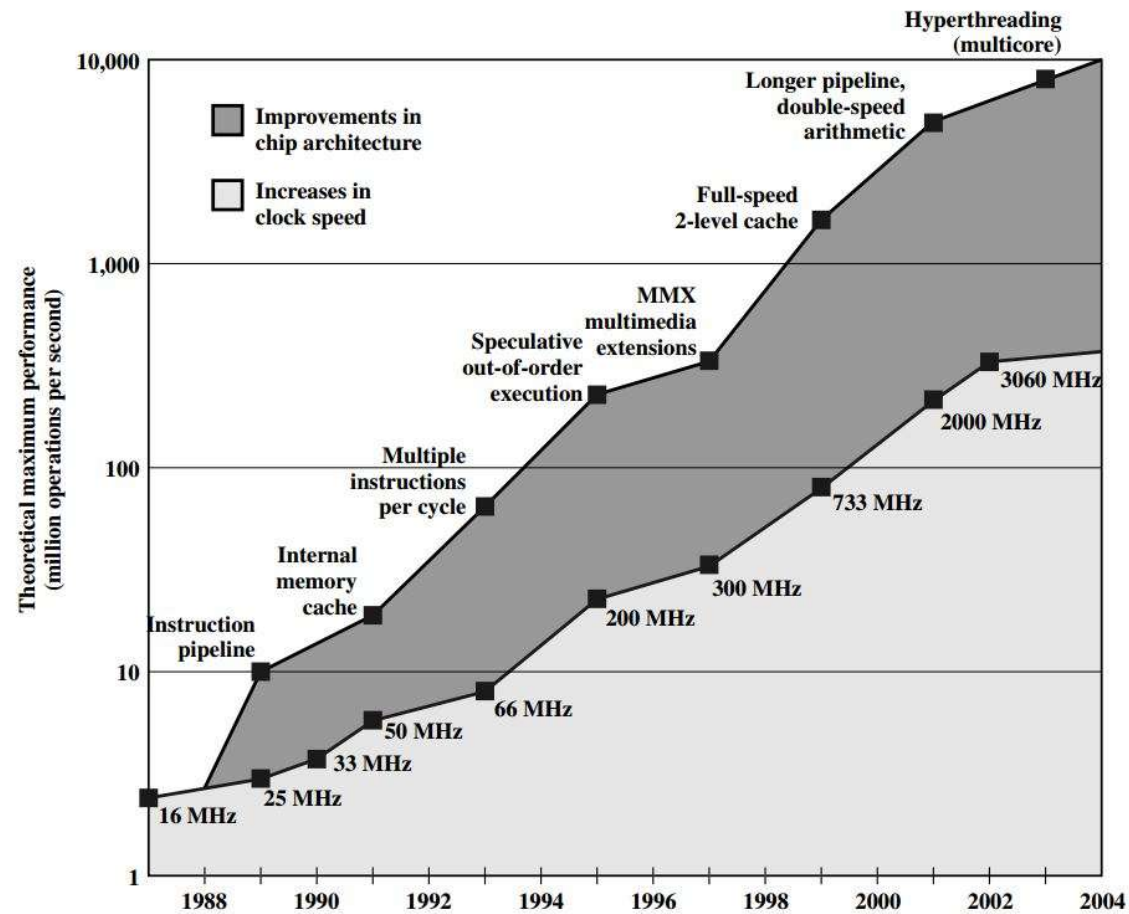
- 100,000 - 100,000,000 komponen pada satu chip

Ultra large  
scale  
integration -  
kedepan

- Over 100,000,000 komponen pada satu chip



# Intel Microprocessor Performance



# Perbedaan Kecepatan

- Terjadi ketidak seimbangan antara kecepatan prosesor dan komponen yang lain, processor semakin melaju ke GHz sementara memory dan I/O relatif stagnan kecepatannya.

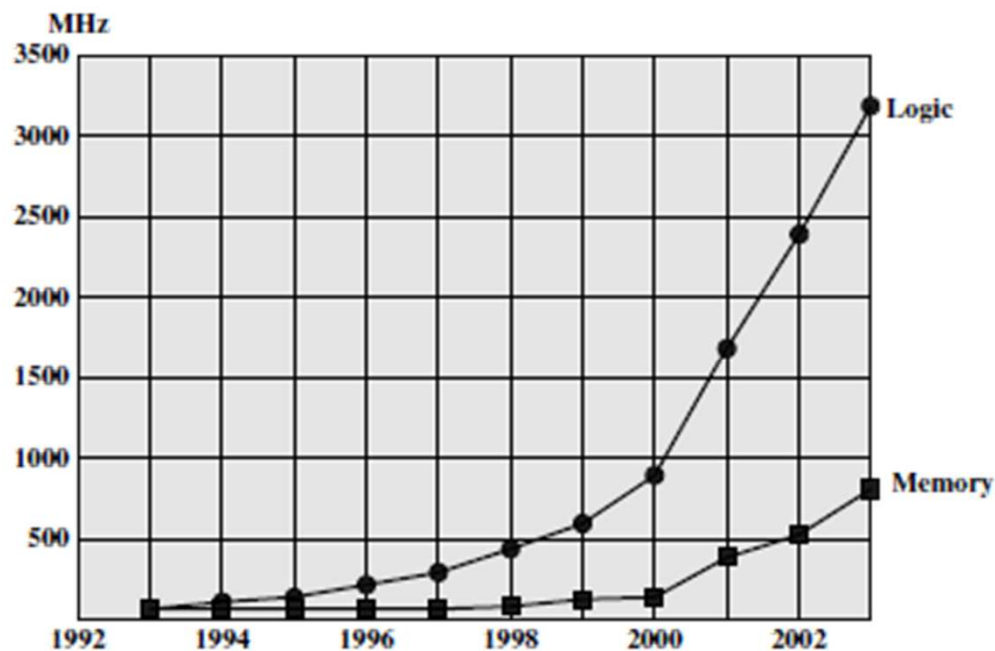


Figure 2.10 Logic and Memory Performance Gap [BORK03]

# Kecepatan I/O

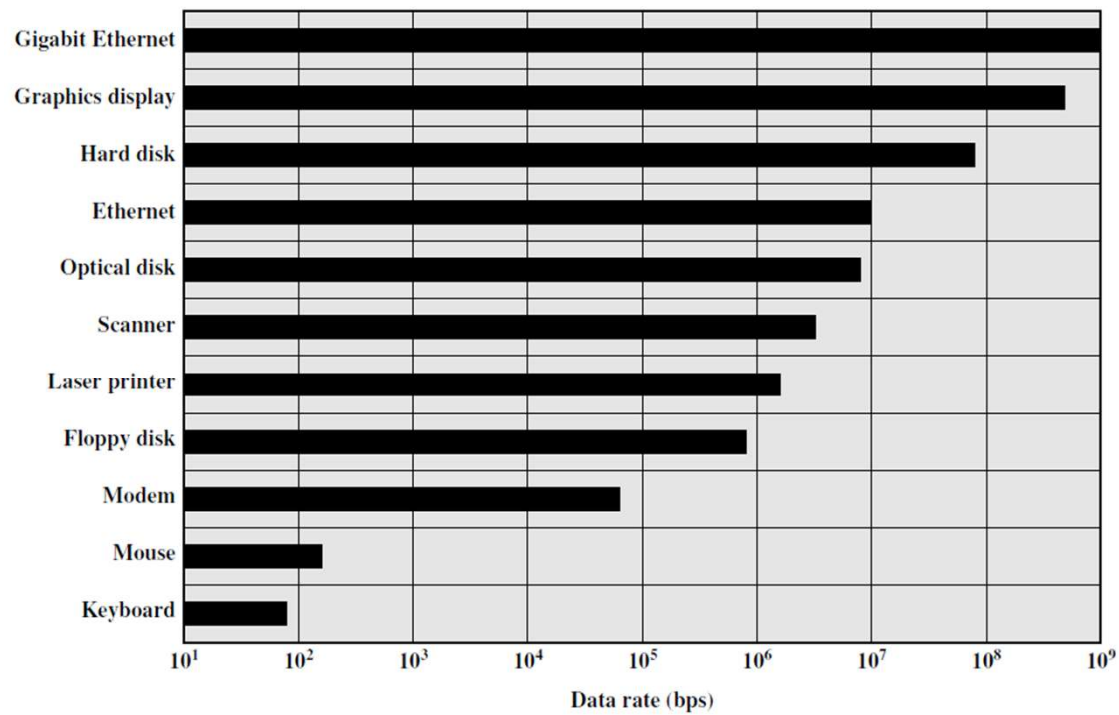
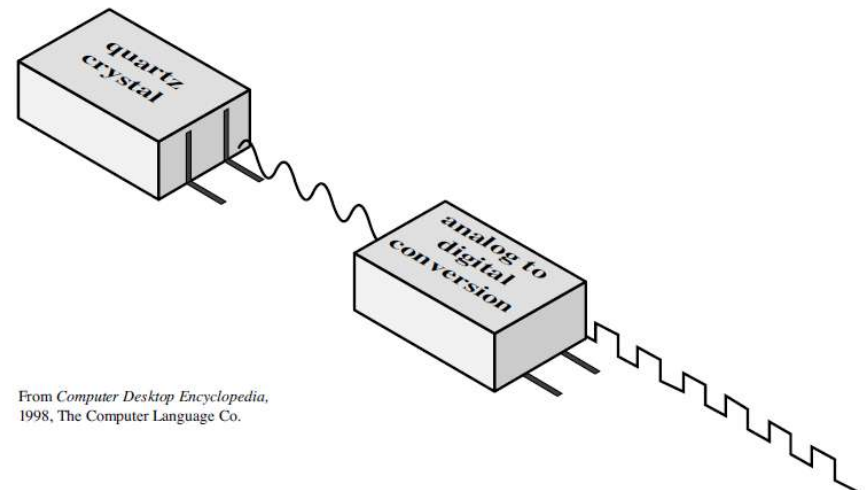


Figure 2.11 Typical I/O Device Data Rates

# Clock

- Clock adalah komponen pemacu / penyerempak semua komponen dalam sistem komputer
- Clock dihasilkan lewat komponen clock osilator untuk menjamin kestabilan frekuensi yang dihasilkannya.



From *Computer Desktop Encyclopedia*,  
1998, The Computer Language Co.

Figure 2.14 System Clock

# Laju Eksekusi Instruksi

- Prosesor yang di pacu oleh clock dengan frekuensi tetap  $f$ , mempunyai waktu siklus tetap  $t$  dengan  $t = \frac{1}{f}$
- Jika jumlah baris program yang harus dieksekusi  $I_c$  dan rata-rata siklus per intruksi CPI
- Maka waktu yang dibutuhkan untuk eksekusi sebuah program adalah  $T = I_c \times CPI \times t$
- Ukuran umum yang digunakan untuk mengukur performansi CPU adalah

$$MIPS\ rate = \frac{I_c}{T \times 10^6} = \frac{f}{(CPI \times 10^6)}$$

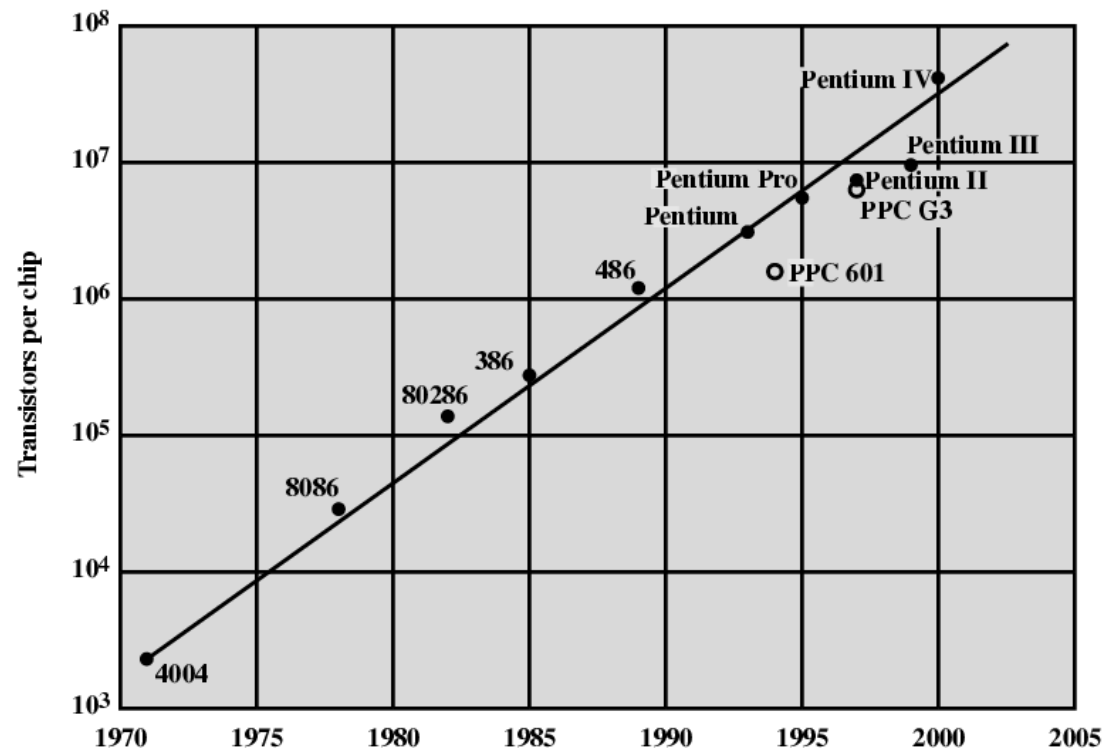
Contoh kasus : Prosesor 400 MHz

- Rata-rata CPI =  $(1 \times 0.6) + (2 \times 0.18) + (4 \times 0.12) + (8 \times 0.1)$   
= 2.24

- Rata-rata MIPS =  $(400 \times 10^6) / (2.24 \times 10^6) = 178$

Instruction Type	CPI	Instruction Mix
Arithmetic and logic	1	60%
Load/store with cache hit	2	18%
Branch	4	12%
Memory reference with cache miss	8	10%

# Perkembangan Jumlah Transistor dalam CPU

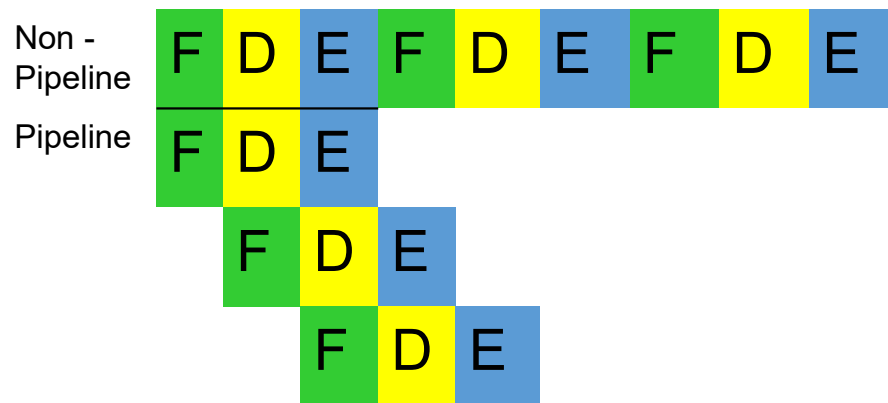


# Instruksi

- Instruksi  
ADD AX,BX
- $\mu$ Intruksi  
Buka saluran ALU 1
- $\mu$ program  
Urutan  $\mu$ instruksi yang merupakan hasil decode dari intruksi



# Pipeline



- Proses antrian intruksi untuk mempercepat komputasi CPU dengan melakukan proses paralel untuk siklus FDE
- AMD berhasil lebih cepat dari Intel dengan menggunakan lebih sedikit pipeline. AMD melakukan analisa kecenderungan program

# Cache

Terdapat 2 atau 3 tingkat (L1, L2, L3)



Mengantisipasi dgn mengambil instruksi-intruksi yang diperkirakan akan dijalankan selanjutnya



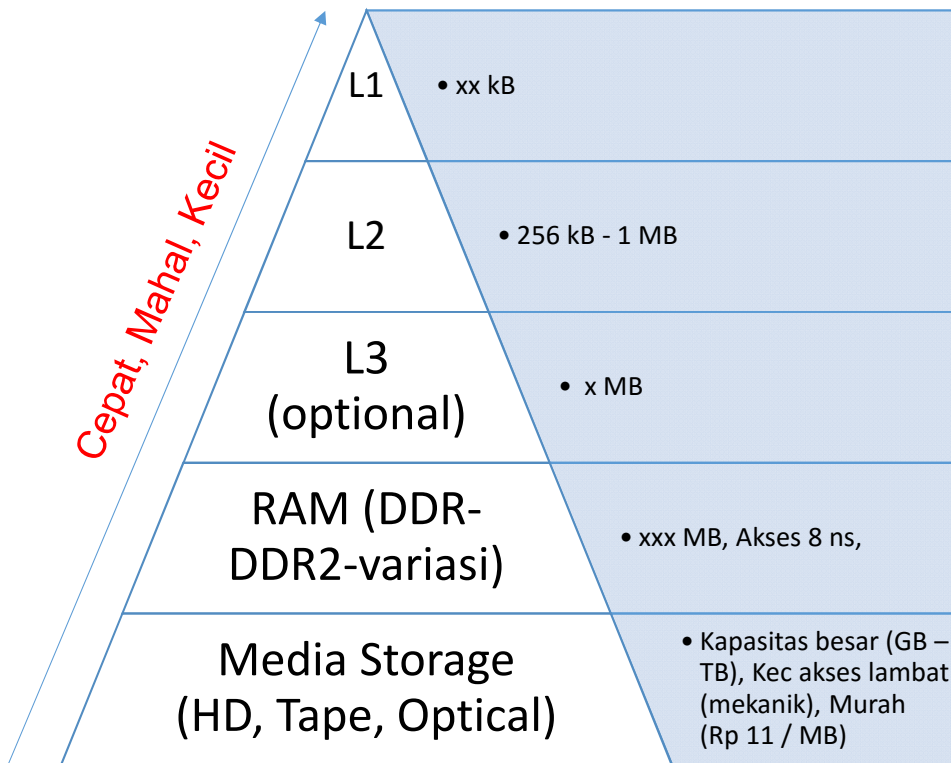
Kecepatan  $L1 > L2 > L3 > RAM$  (1 ns, 5 ns, 10 ns, 40 ns)



Kapasitas  $L1 < L2 < L3 \ll RAM$  (xx kB, xxx kB, x MB, xxxx MB)



# Piramida Memori



Jenis	MB/\$	Gbps
DDR3		68
DDR2	30	51.2
DDR1	17	51.2
SDR	7	xx
DVD	1000	0.172
HD	2500	3
Flash	150	0.5

# L1 on board & on chip

L1 mulai ada di 386 dalam bentuk 8 sd 32 kB on board (diletakan di motherboard) menggunakan cache controller khusus dr Intel

L1 mulai on chip di 486 (386+387+L1) sebesar 8 kB (4 kB data & 4 kB code), L2 on board

L2 on chip mulai sejak Pentium

# Cara Peningkatan Performansi

1. Cache - Kecenderungan program diakses secara berurutan, maka daripada pada satu siklus hanya mengambil 1 instruksi akan lebih “baik” jika juga mengambil intruksi-instruksi selanjutnya..

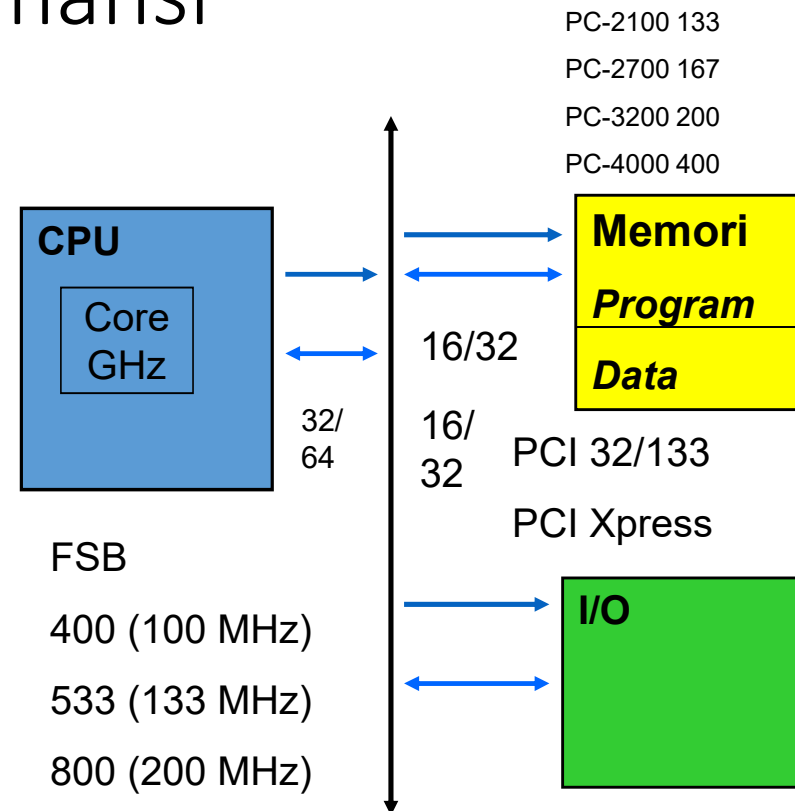
**Cache Hit** → data ada dicache

**Cache Miss** → data tidak ada dicache sehingga harus diambil dari memori utama

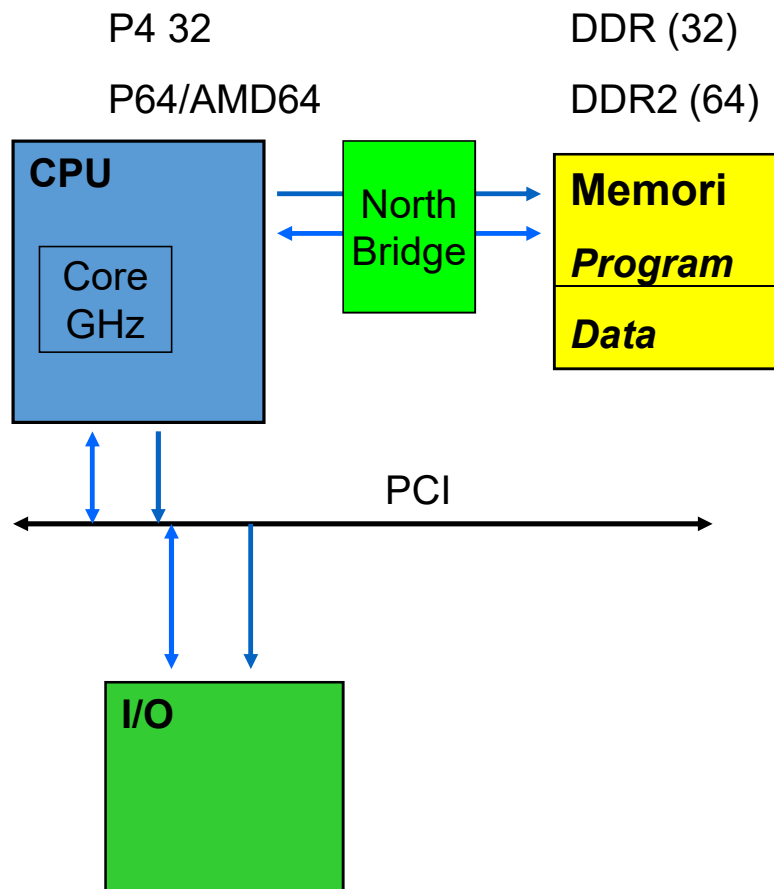
Target : **Cache Miss seminimal mungkin** (contoh cache miss 5 % sehingga cache hit 95%) →  
Diperlukan **algoritma cache yang efisien** (write through, write back dll) sesuai dengan kecenderungan program

# Cara Peningkatan Performansi

2. **Bottleneck** di BUS (solusi diciptakan bus selebar bus data & kontroller (Northbridge) khusus antara CPU dan Memori)



# High Speed Memory Bus



- Berdasarkan skema sebelumnya, skema berikut merupakan solusi pemisahan jalur BUS sehingga dapat diminimalisir bottleneck pada jalur BUS.

# Cara Peningkatan Performansi

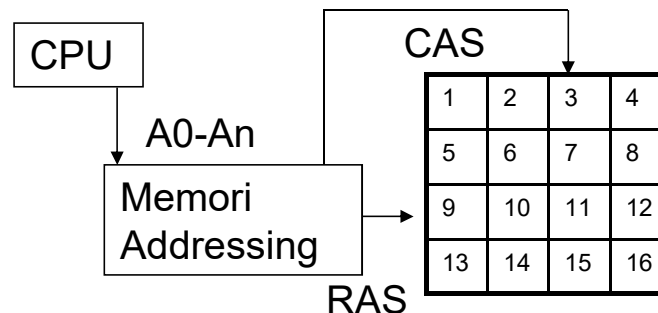
## 3. Kecepatan Akses Memori

- 2 – 3 – 3 – 7 (-1T)
  - CAS Latency 2, 2.5, 3 (DDR) 2,3 (SDR)
  - RAS to CAS Delay 2,3
  - RAS Precharge 2,3
  - Active to Precharge 3,4,5,6,7
  - Command Rate
  - (-1T) Delay antara akses
  - Memory Cyclic Time (Total)

- Kapasitas RAM yang mencapai xx MB sd GB (tepatnya xx Mb sd Gb per chip 1 IC biasanya hanya mempunyai 1 - 4 pin data) memerlukan 'kaki' (pin) yang banyak. Misal : 1 Gb per IC membutuhkan 30 pin alamat (ukuran IC jadi besar, tidak diinginkan). Solusi : Pengalamatan dibagi menjadi kolom & baris

Butuh 16 jalur untuk mengisi semua sel

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----



+ Butuh 8 jalur

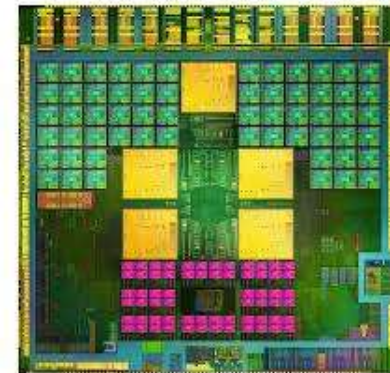
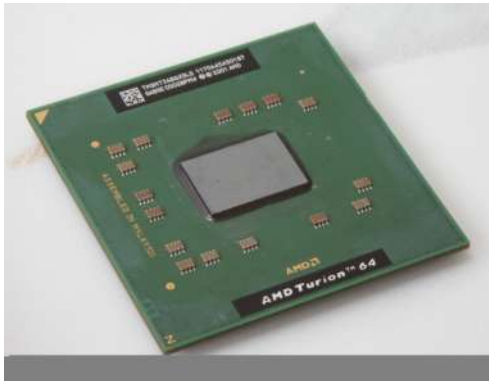
- Pengalamatan lebih kompleks (baris & kolom)



# Cara Peningkatan Performansi

## 4. Multi-prosesor

- Dimulai dari generasi intel Dual Core sampai saat ini Multi Core dalam rangka meningkatkan kecepatan tanpa harus merancang chip prosesor yang rumit



# Hukum Amdahl (Gene Amdahl 1967)

$$\text{Speedup} = \frac{\text{waktu eksekusi program di prosesor tunggal}}{\text{waktu eksekusi program di N prosesor paralel}}$$

$$\text{Speedup} = \frac{T(1 - f) + Tf}{T(1 - f) + \frac{Tf}{N}}$$

$$\text{Speedup} = \frac{1}{(1 - f) + \frac{f}{N}}$$

f = bagian program yang paralel di prosesor tunggal

N = jumlah prosesor