BUS, Cache & Shared Memory

Team Dosen

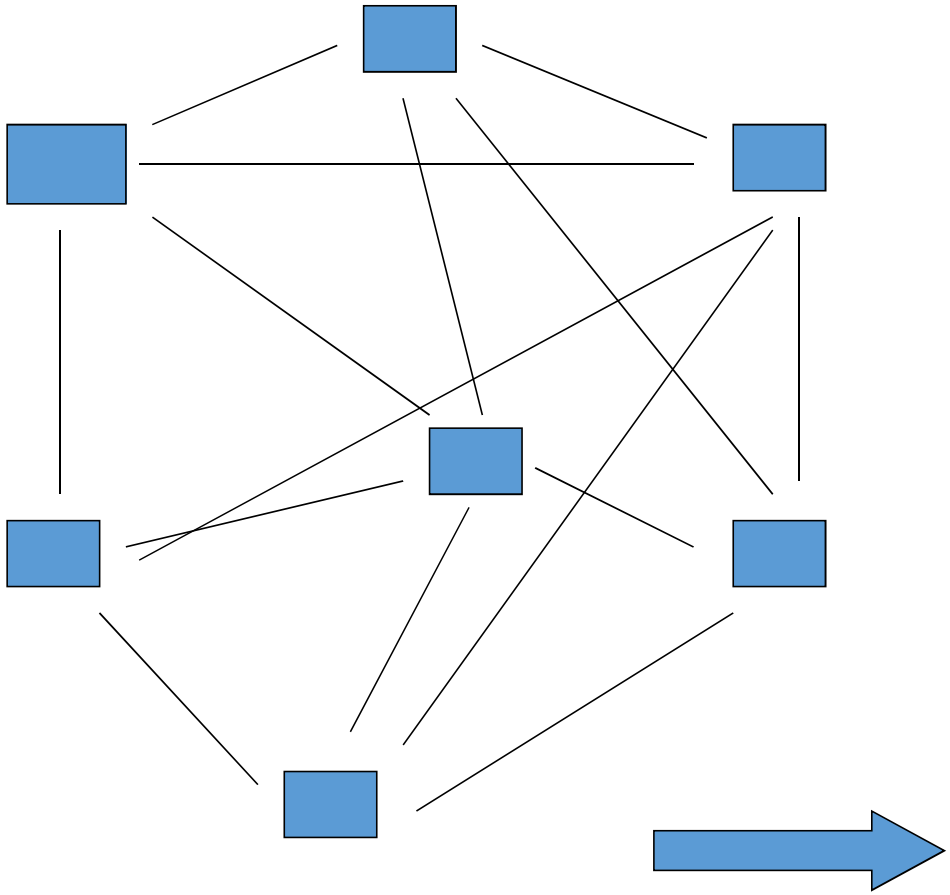
Telkom University

2016

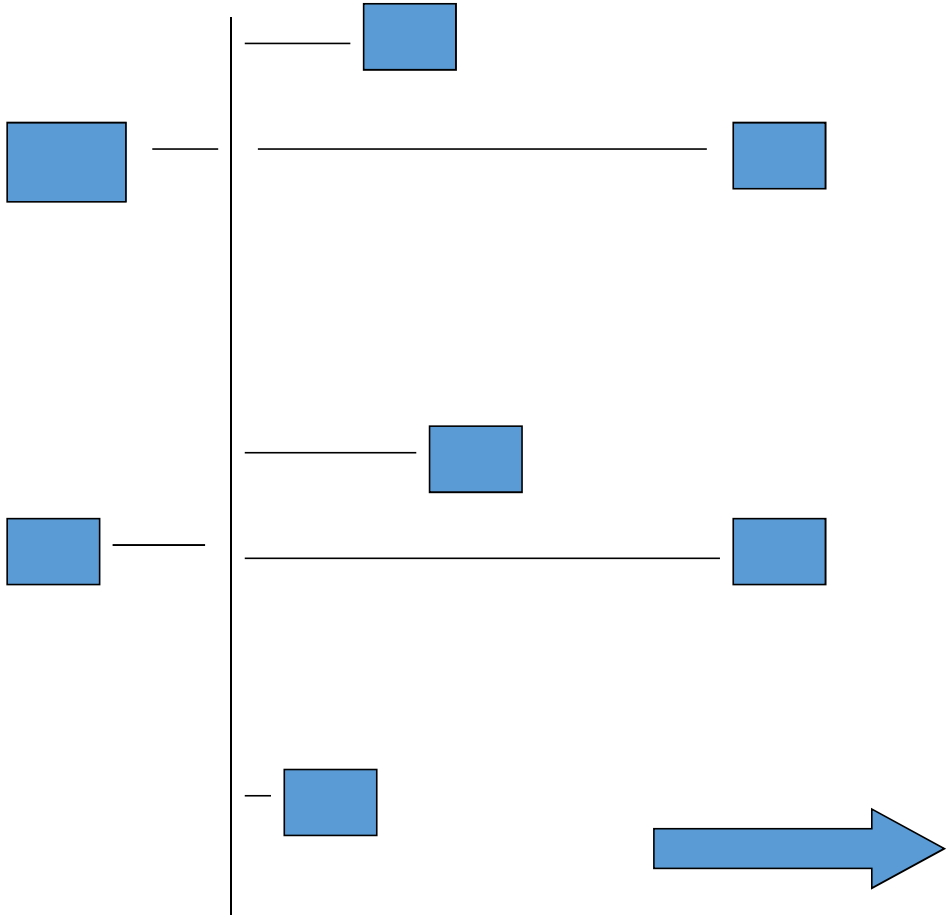
Bus ?  ?

- Jalur komunikasi antar devais
- Bersifat broadcast
- Hanya satu divais yang bisa mengirim data pada satu saat
- Biasanya merupakan kelompok fungsional
  - Jumlah kanal di suatu bus
  - Contoh : 32 bit data bus adalah 32 kanal data masing-masing kanal satu bit

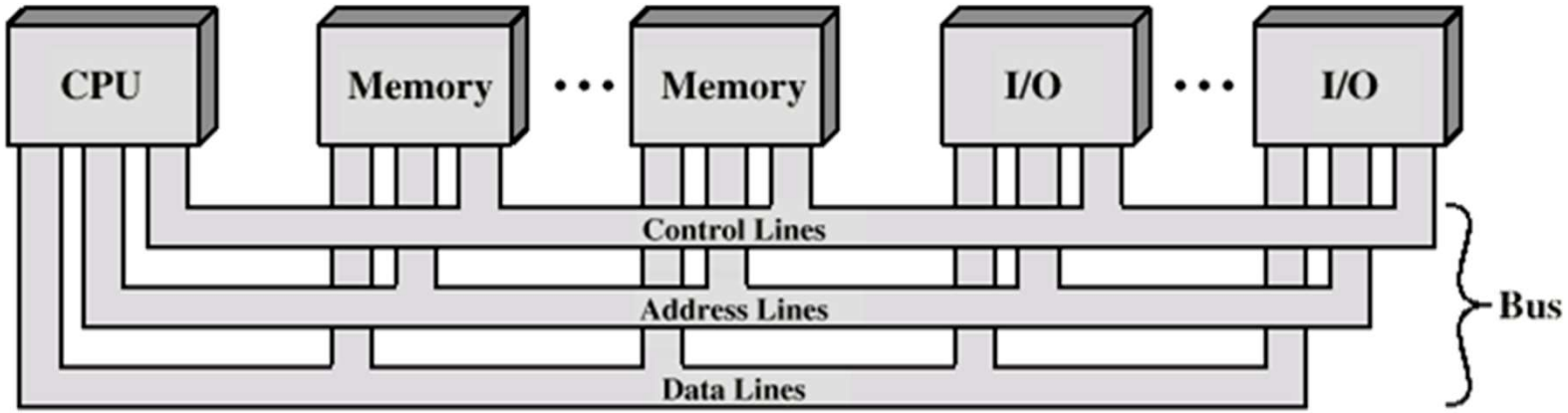
Sebelum BUS



Ide Dasar BUS



Skema Interkoneksi Bus



Data Lines

- Mendukung jalur untuk memindahkan/mempertukarkan data.
- Disebut juga Bus Data.
- Biasanya terdiri dari 32/64/128 atau lebih jalur.
- Disebabkan hanya bisa mendukung 1 bit pada saat yang bersamaan, maka jumlah baris yang menunjukkan berapa bit yang dapat ditransfer tiap waktunya.
- Lebar bus data ini adalah hal yang paling penting untuk menaikkan performansi komputer.
- Bayangkan jalur yang dimiliki sebuah bus data adalah sebesar 32 bit, sedangkan data yang akan ditransfer adalah sebesar 64 bit. Maka dibutuhkan 2 cylce untuk menyelesaikan transfer data.

Bus Data

- Membawa sinyal informasi
  - Membawa intruksi dan data
- Lebar bus menentukan performa
  - XT 8bit/8088 (1981),
  - ISA-16/80286 (1984),
  - EISA 32 bit (1986),
  - PCI 32/80386(1986),
  - PCI-64/PCI-Express 64/AMD64 (2003)
  - Semakin lebar bus data akan semakin ‘powerfull’ sistem komputer tersebut, kemampuan akses data semakin banyak pada satu saat

Address Lines

- Digunakan untuk menentukan siapa pengirim dan penerima data yang dilalui bus data
- Apabila sebuah prosesor ingin membaca word (sebesar 8, 16, atau 32 bit) data dari memori, maka prosesor akan meletakkan alamat dari word yang dimaksud ke address lines.
- Biasanya, lebar bus alamat ini menentukan sebesar apa sebuah memori yang dimiliki oleh sebuah sistem.
- Address lines umumnya digunakan juga untuk menjadi jalur alamat untuk I/O.
- Sebagai contoh : 8 bit bus alamat, 01111111 ke bawah digunakan sebagai pengalamatan ke modul memori, sedangkan 10000000 ke atas digunakan sebagai pengalamatan ke modul I/O

Bus Alamat

- Buat uP semua divais adalah kumpulan alamat
- Menunjukan lokasi dari memori/devais
- Lebar bus menentukan besarnya ruang memori yang bisa diakses
  - 8080 mempunyai bus alamat 16 bit yang berarti mempunyai ruang alamat 64k
  - 8088 mempunyai bus alamat 20 bit (A0 sd A19) yang berarti mempunyai ruang alamat 1M
  - DDR **256 MB** = 64 Mb x **4** (D0 sd D3) x 8 IC, lebar bus data 32 bit (D0 sd D31)
- Sejarah
  - 1981 – 8088/8086 → 1 MB (umumnya RAM 640 kB) -- 20 bit bus alamat
  - 1984 – 80286 → 16 MB (umumnya 2 MB) – 24 bit bus alamat
  - 1987 – 80386 → 4 GB (umumnya 4 MB) – 32 bit bus alamat
  - 2000 – P4 → 64 GB (umumnya 1 GB) – 36 bit bus alamat
  - 2014 – Core i7 → 36 bit bus alamat

Control Lines

- Digunakan sebagai kontrol akses terhadap data dan address lines. Hal ini dikarenakan data dan adress lines merupakan sesuatu yang dibagi kepada semua komponen yang ada.
- Maka harus ada yang dapat mengkontrol penggunaannya.
- Kontrol melakukan transmisi sinyal command dan timing.
- Timing mencatat validitas dari sebuah data, command memberikan perintah operasi yang harus dilakukan.

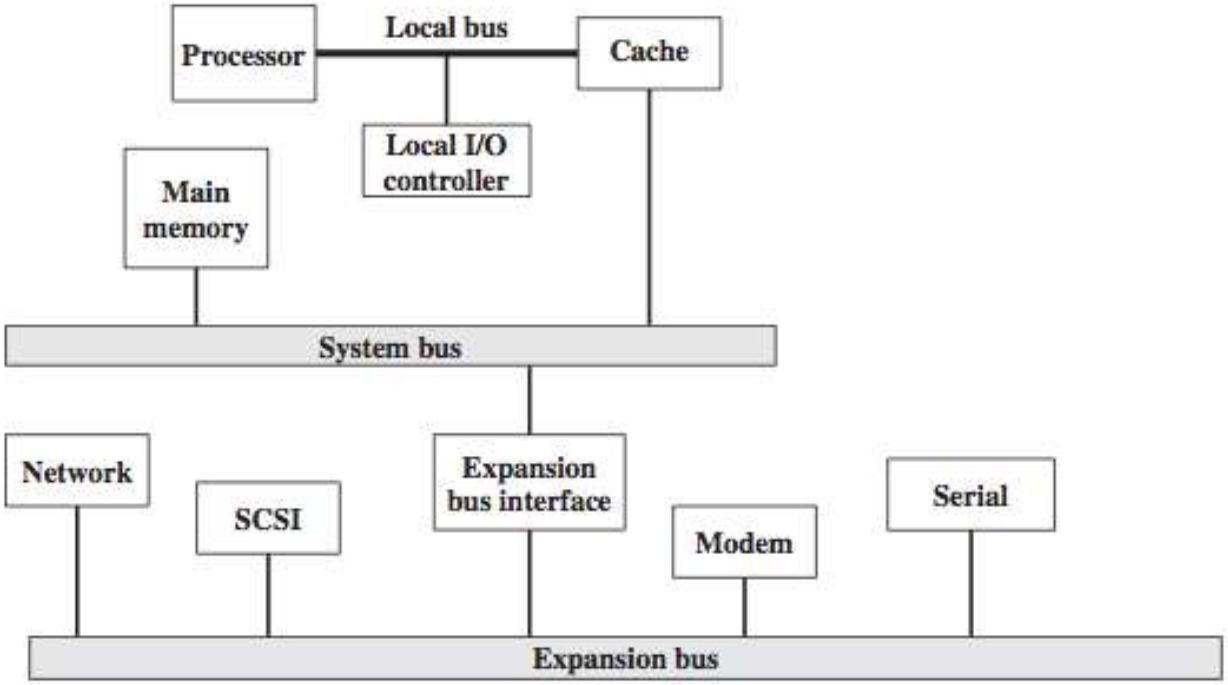
Bus Kendali

- Informasi Kendali dan Timing
  - Sinyal baca/tulis memori (MWrite, MRead)
  - Sinyal baca/tulis I/O (IOWrite, IORead)
  - Transfer ACK
  - Bus request
  - Bus grant
  - Kendali Kanal DMA
  - Interrupt request (IRQn)
  - Interrupt ACK
  - Sinyal Clock
  - Reset

Multiple-Bus Hierarchies

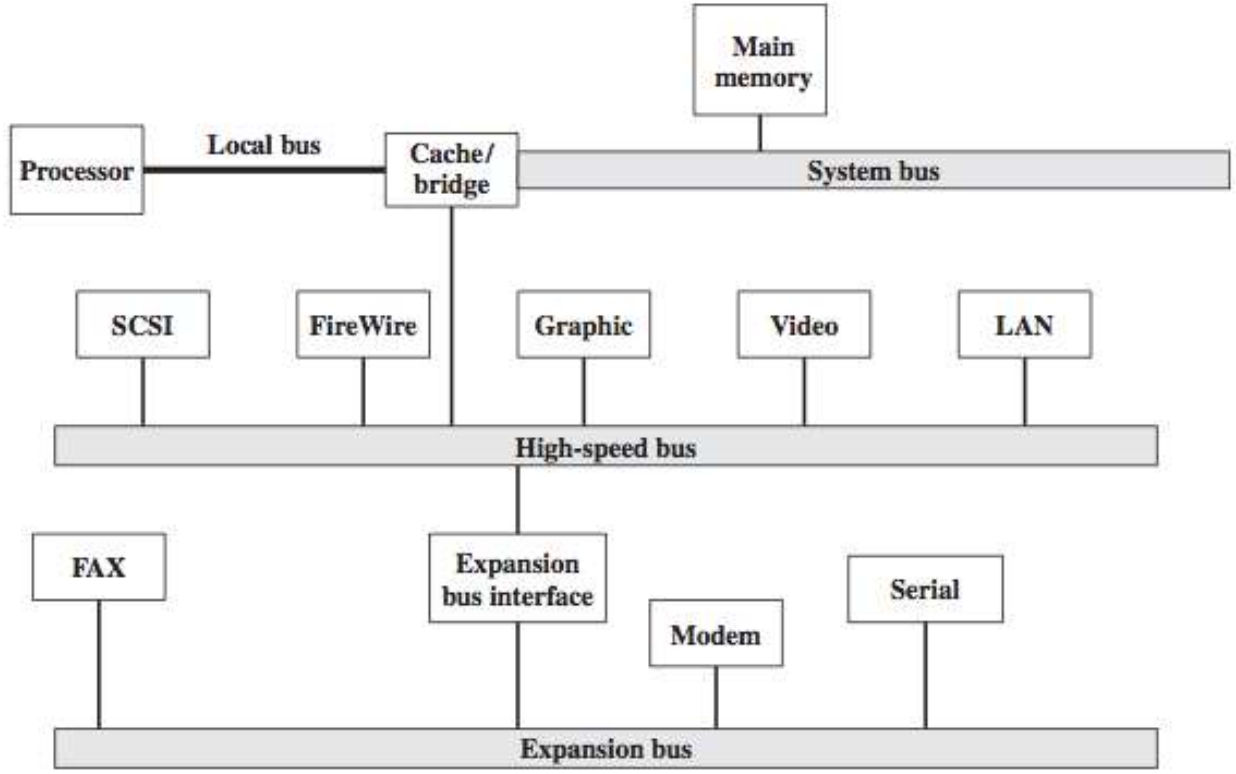
- Semakin banyak device yang bisa terhubung dengan bus, maka performance sistem akan semakin menurun.
  - Pada umumnya, semakin banyak perangkat yang terhubung dengan bus, menyebabkan lebar bus akan semakin besar, dan menyebabkan delay propagasi juga semakin besar. Delay ini menyebabkan waktu yang dibutuhkan untuk perangkat melakukan koordinasi dengan bus meningkat. Delay inilah yang dapat berpengaruh ke performansi.
  - Bus bisa saja menjadi bottleneck, berdasarkan semakin besar data yang ditransfer dibandingkan dengan lebar bus yang ada. Dapat diatasi dengan mempercepat bus rate dan melebarkan bus yang ada.

Bus (Tradisional)



(a) Traditional bus architecture

Bus (High Performance)



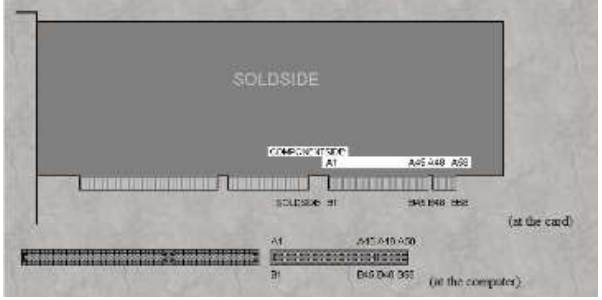
(b) High-performance architecture

Bus Video

- Pada dasarnya video card membutuhkan bus data yang cepat, bus alamat relatif sedikit (umumnya video card hanya butuh alamat sebesar 64 kB)
- PC 8 bit – 1981
- ISA 16 bit – 1984
- VESA 32 bit – 1988
- PCI 32 bit – 1990
- AGP 32? bit – 1994 → bus khusus video
- PCIe x16 – 2002 (serial bus) → bus khusus video

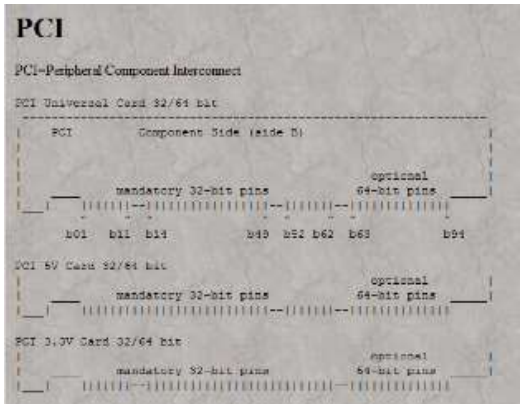
VESA LocalBus (VLB)

VLB=VESA Local Bus
VESA=Video Electronics Standards Association

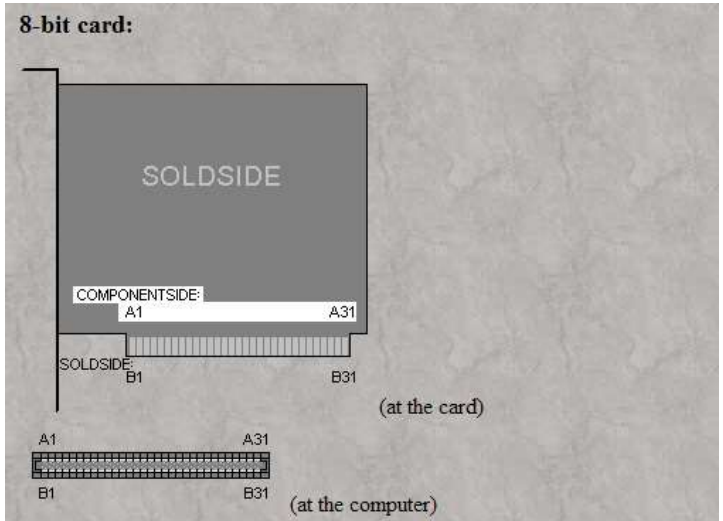


PCI

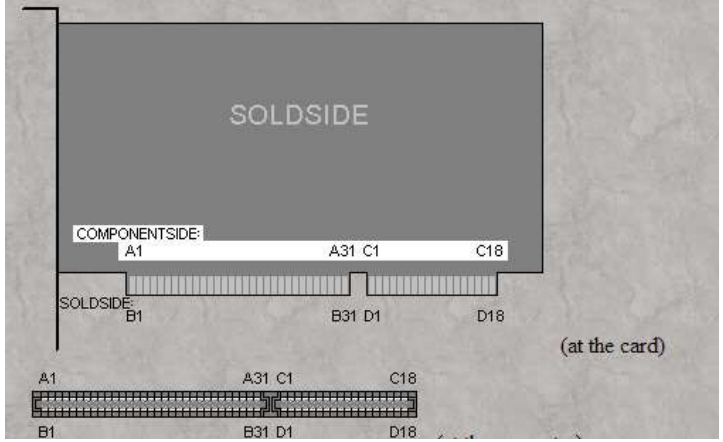
PCI=Peripheral Component Interconnect



8-bit card:



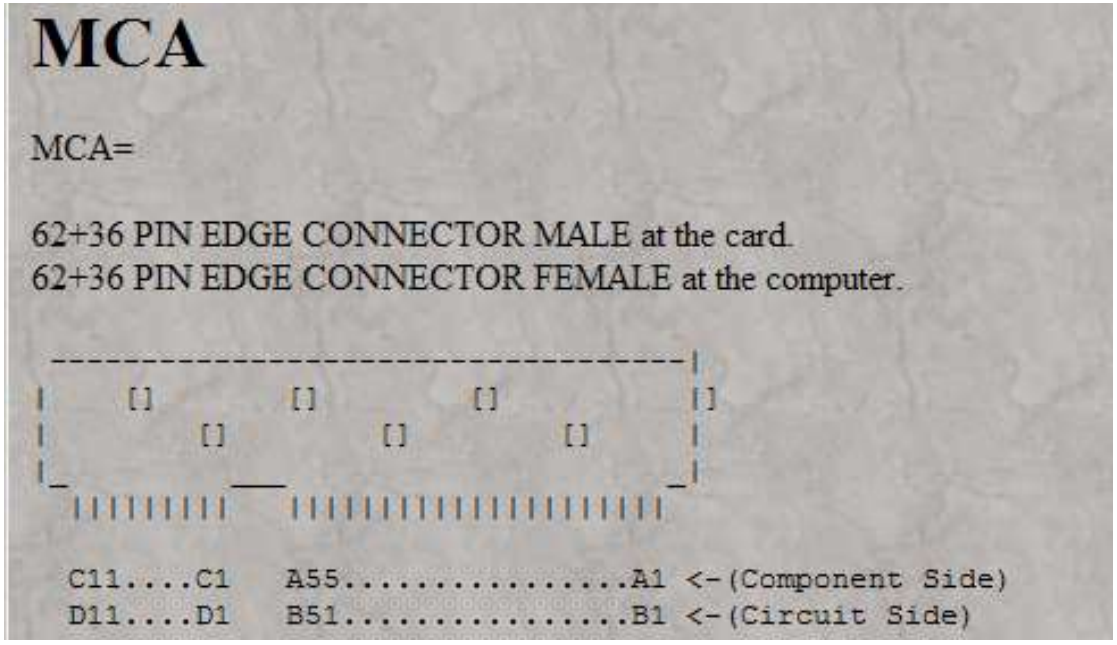
16-bit card:



Telkom

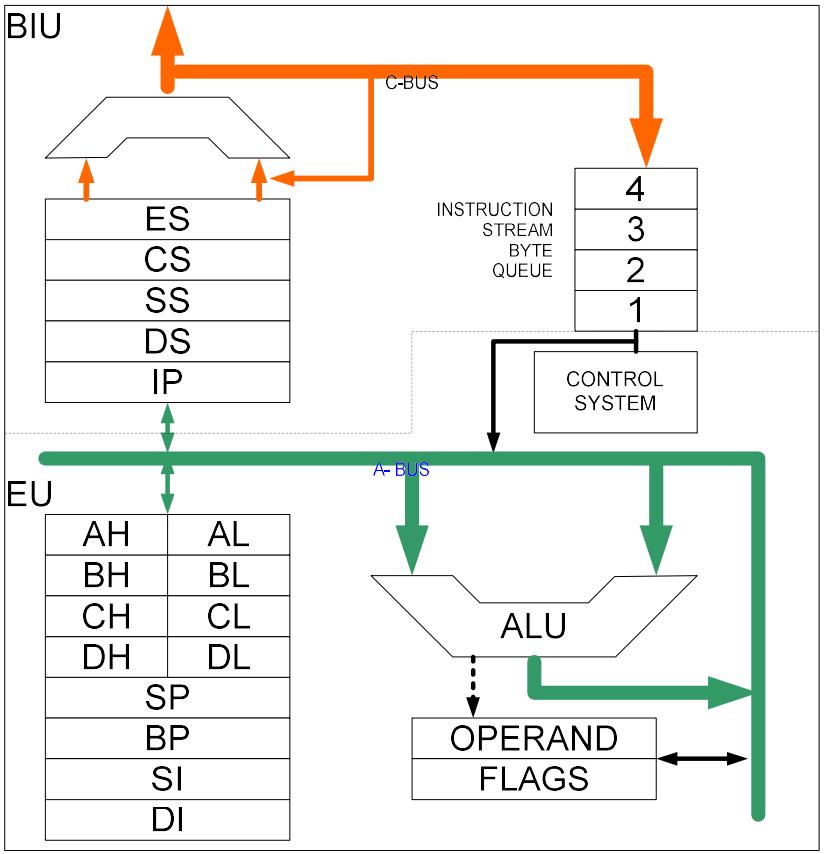
BUS / Slot (Bus + Catudaya)

- PC Bus
- VME
- S100
- DecBUS
- dll



Bus Internal

- Menghubungkan ALU, Register dan komponen lain dalam CPU
- Lebar sesuai lebar register
- Sangat cepat (bekerja di core clock)



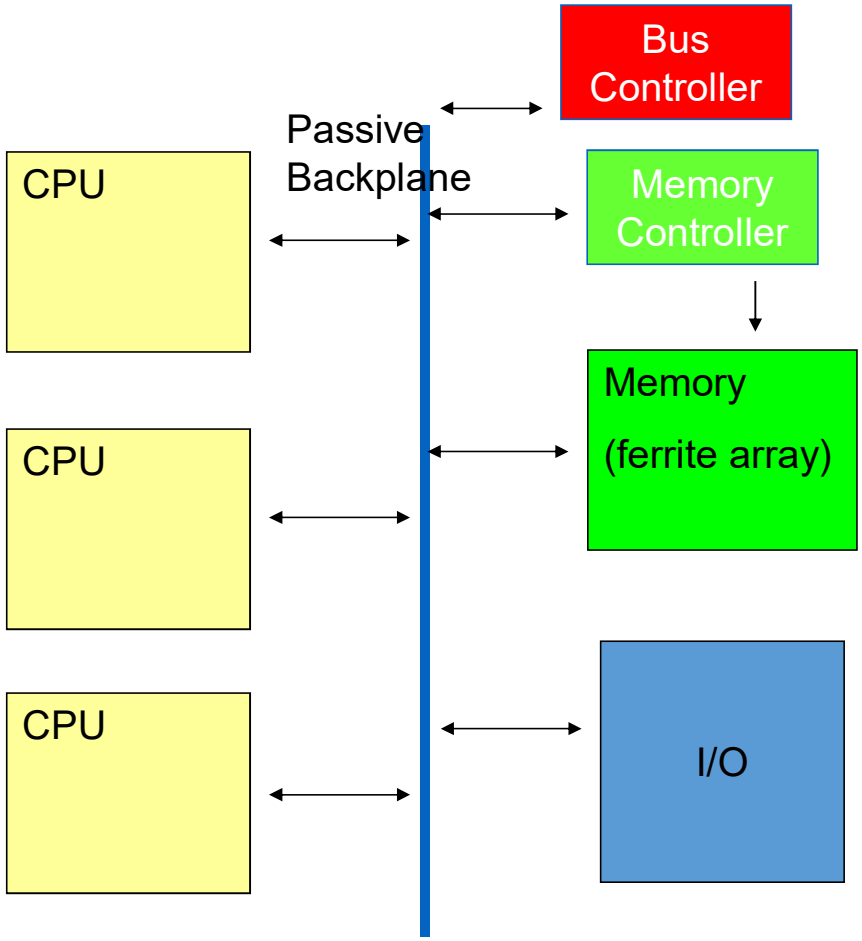
PC Bus

- Bus Memory (High Speed Bus)
  - Lebar sesuai dengan lebar bus external CPU dan lebar memori
    - *SiS membuat bus Hiperstreaming (double width)*
  - Kecepatan dinyatakan dengan FSB (biasanya kecepatan bus * lebar bus dalam byte, **FSB 800** = 200MHz * 4Byte)
  - Menggunakan Northbridge sebagai pengendali

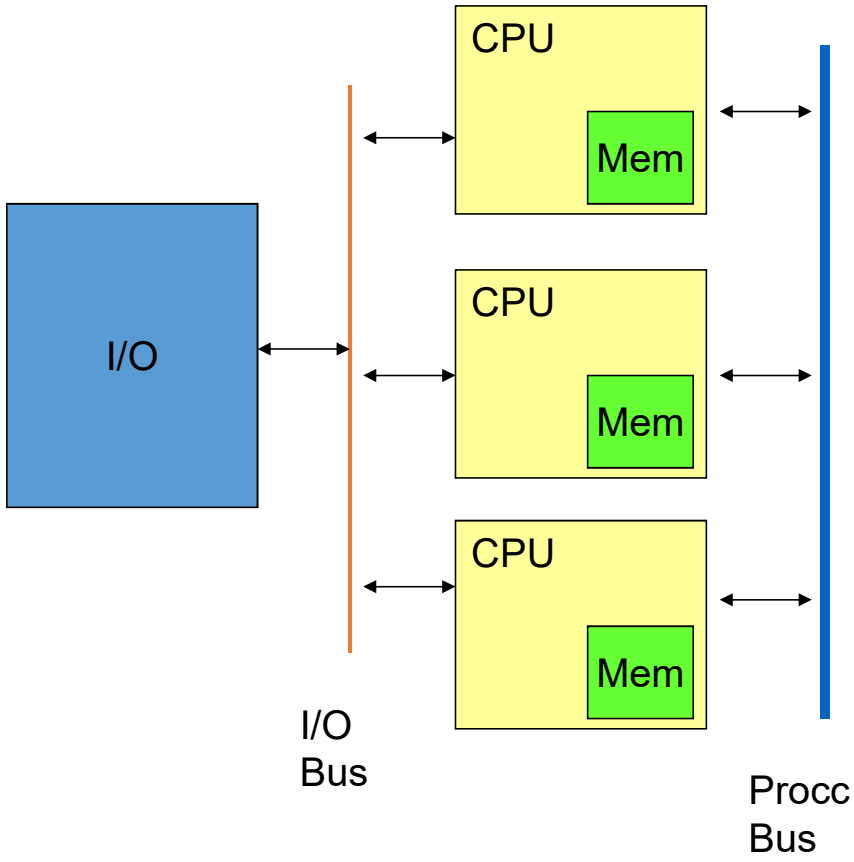
PC Bus

- Expansion Bus
  - XT (8 bit, 4.7 MHz)
  - ISA (16 bit, 8 MHz)
  - EISA (32 bit, 8/16 MHz)
  - Microchannel (IBM PS2 & PowerPC, 32 bit, 16 MHz)
  - VESA (16 bit video, 12 MHz)
  - PCI (32 bit, 33 sd100 MHz)
  - AGP (32 bit video, ….. MHz)
  - PCI Express (32/64 bit, ….. MHz)

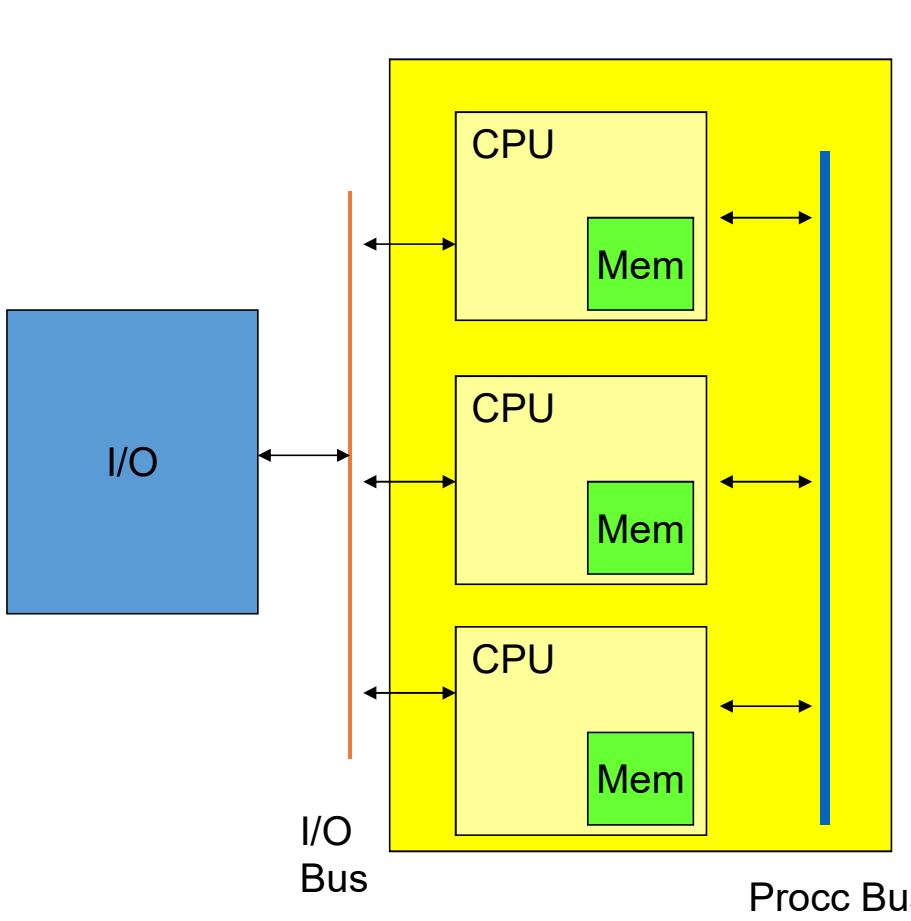
Multiprocessor System (old system)



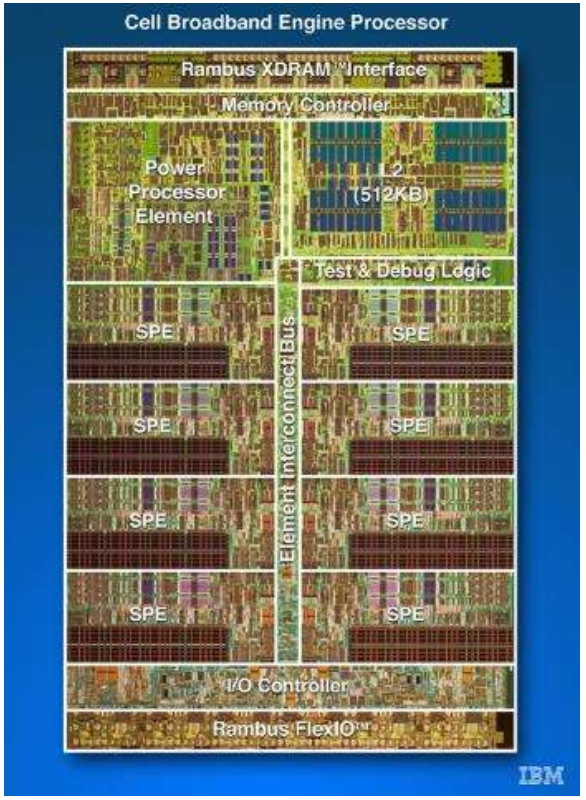
Multiprocessor System (New)



System On Chip (Very New)

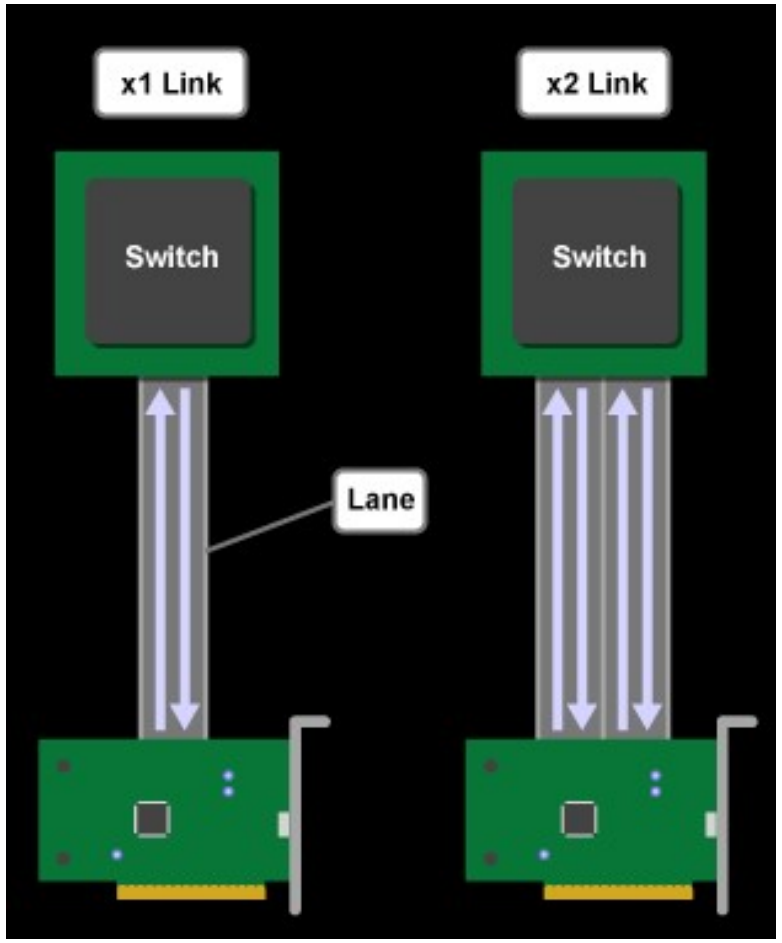


1 chip

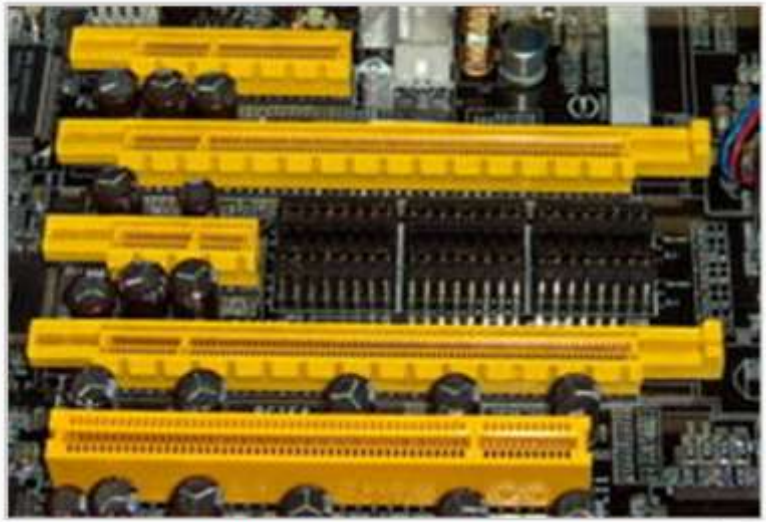


PCIe

- Setiap koneksi di PCIe bisa terdiri dari beberapa jalur serial:
  - Setiap jalur mempunyai lebar 1-bit (4 kabel , setiap pasangan kabel dapat berkecepatan 2.5Gbps)
    - Upstream dan downstream dilakukan secara simultan dan simetris
  - Setiap koneksi dapat terdiri dari 1, 2, 4, 8, 16 jalur (x1, x2, x4, x16)
  - Setiap byte data dikodekan dengan kode 8b/10b , laju data bersih 2 Gbps untuk setiap jalur satu arah.
  - Sehingga, laju data bersih mencapai 250 MBps (x1) 500 MBps (x2), 1GBps (x4), 2 GBps (x8), 4 GBps (x16), each way



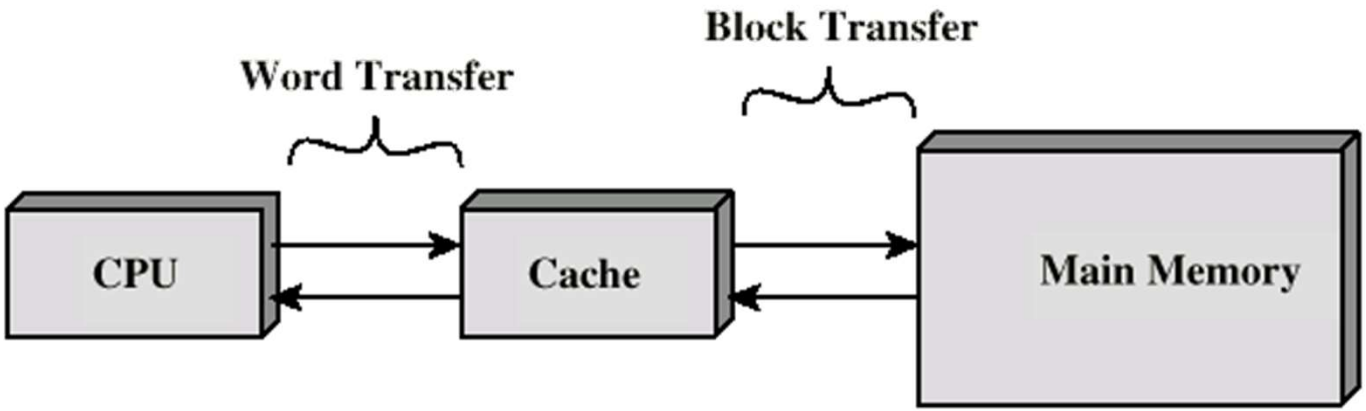
PCI Express Slot



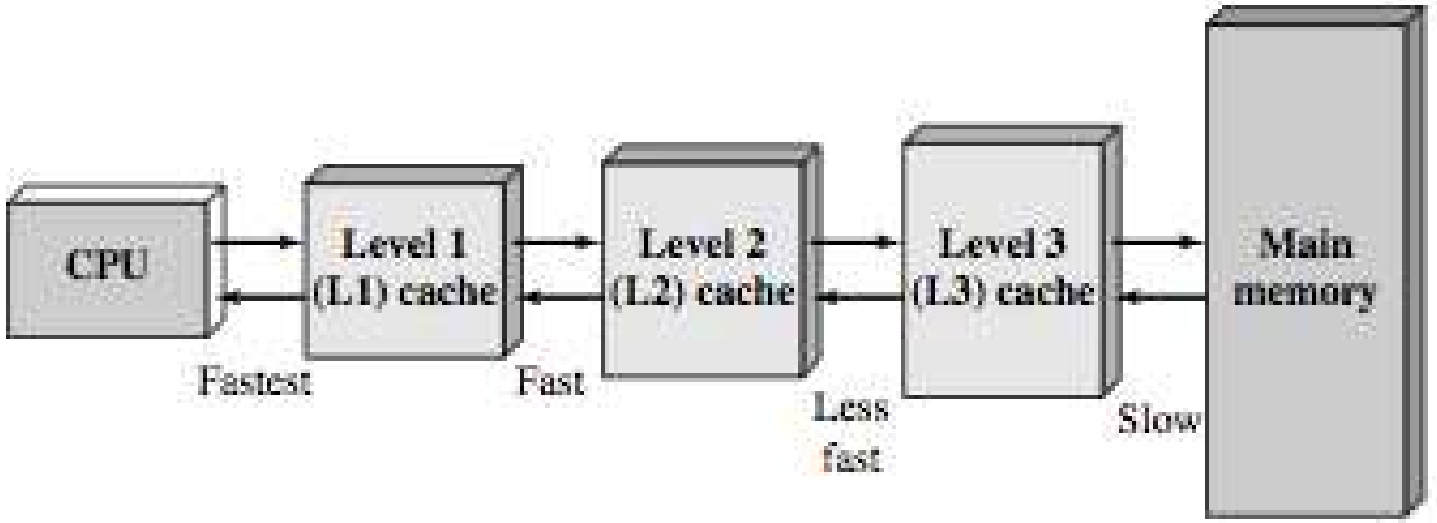
PCI Express slots (from top to bottom: x4, x16, x1 and x16), compared to a traditional 32-bit PCI slot (bottom), as seen on DFI's LanParty nF4 Ultra-D

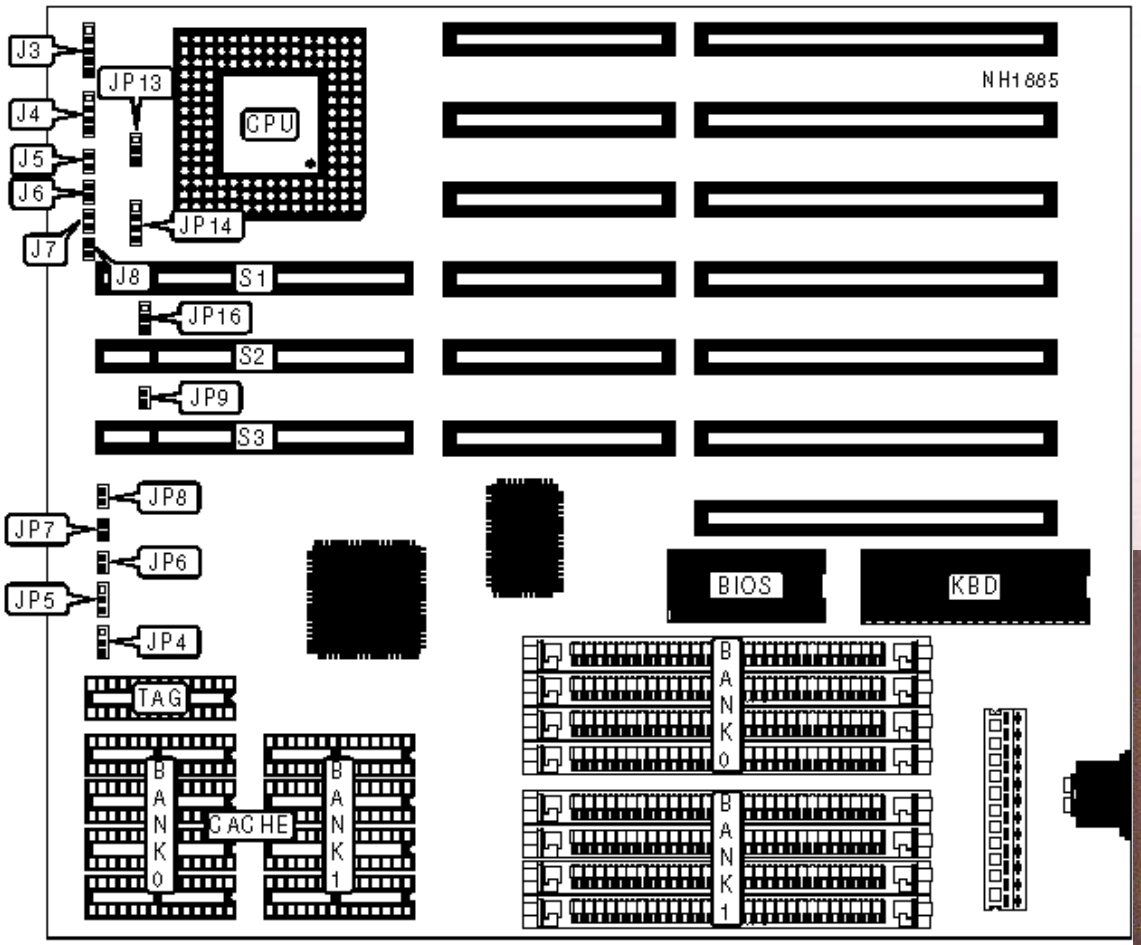
Prinsip Memori Cache

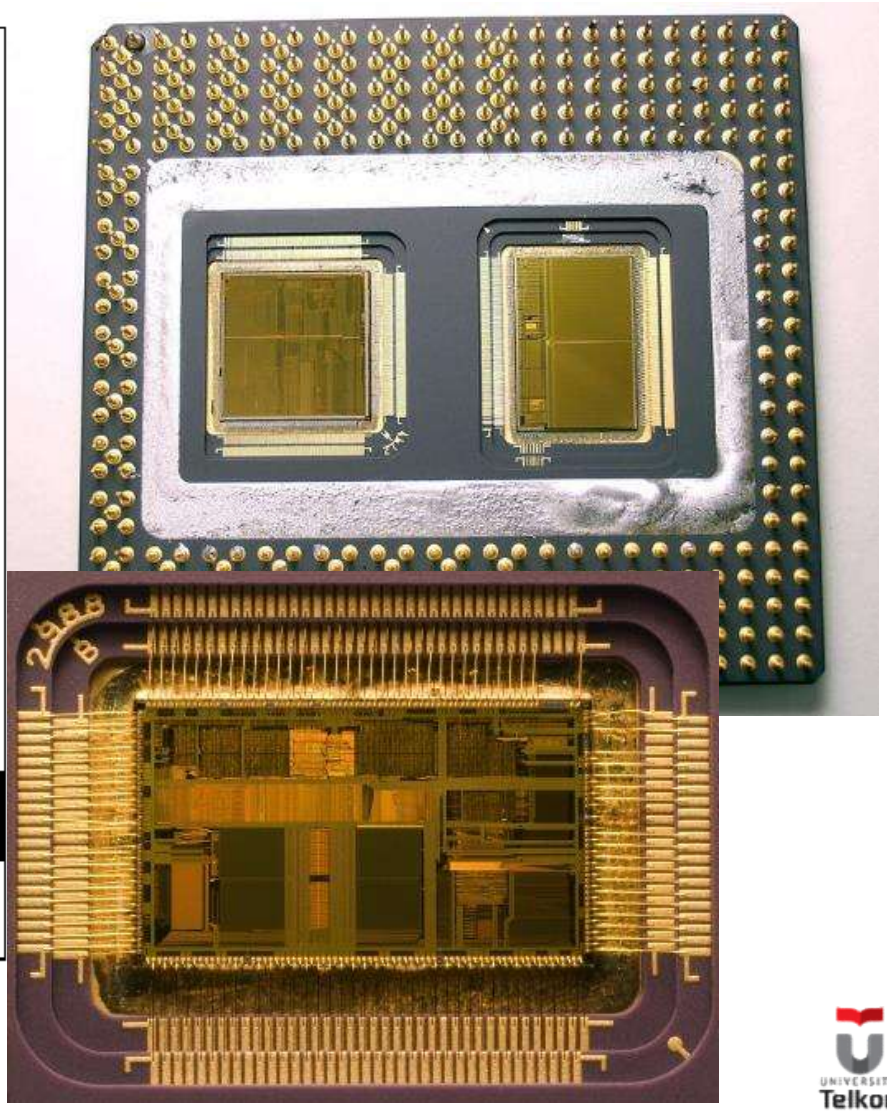
- Memori kecepatan sangat tinggi berkapasitas kecil
- Terletak antara CPU dan memori utama
- Bisa on chip di CPU ataupun di motherboard



MultiLevel Cache

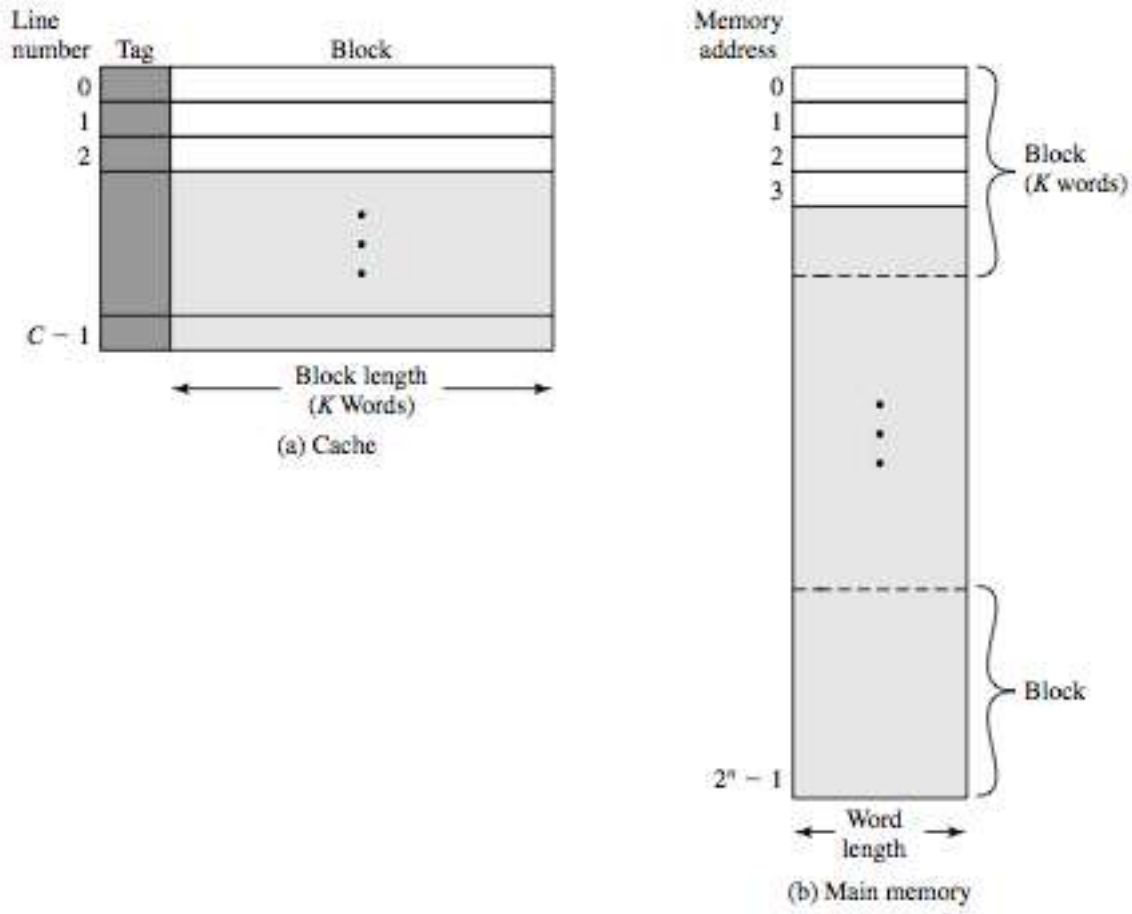






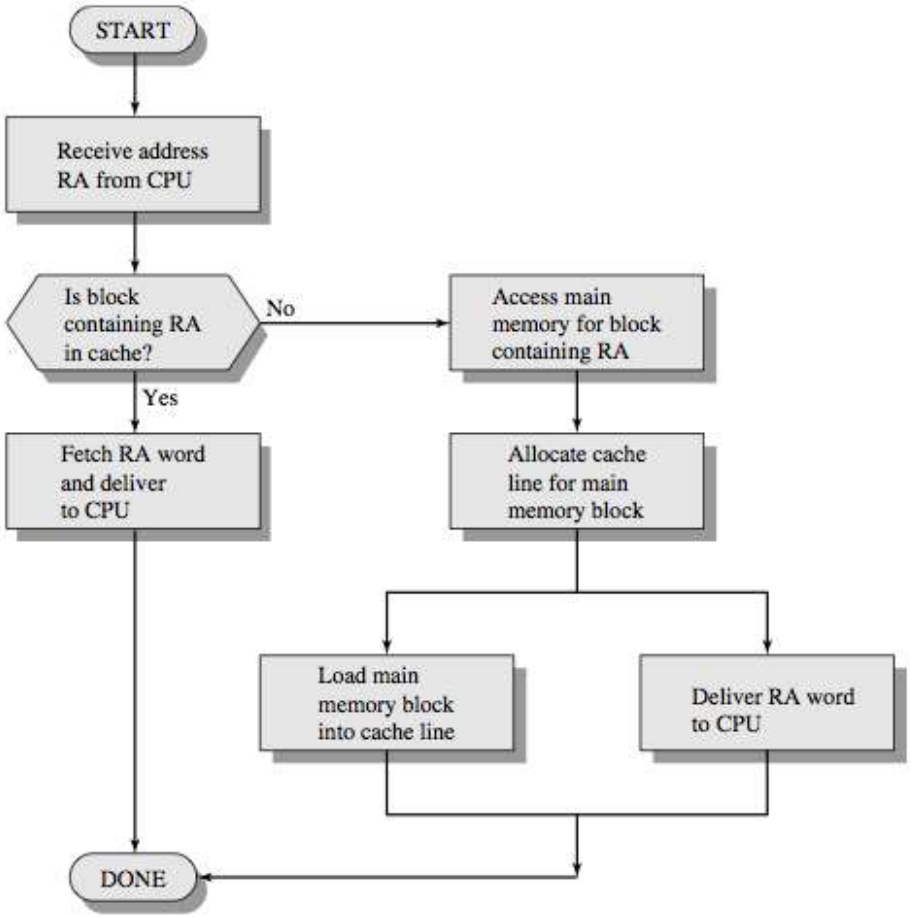
Universitas Telkom

Cache vs Memory

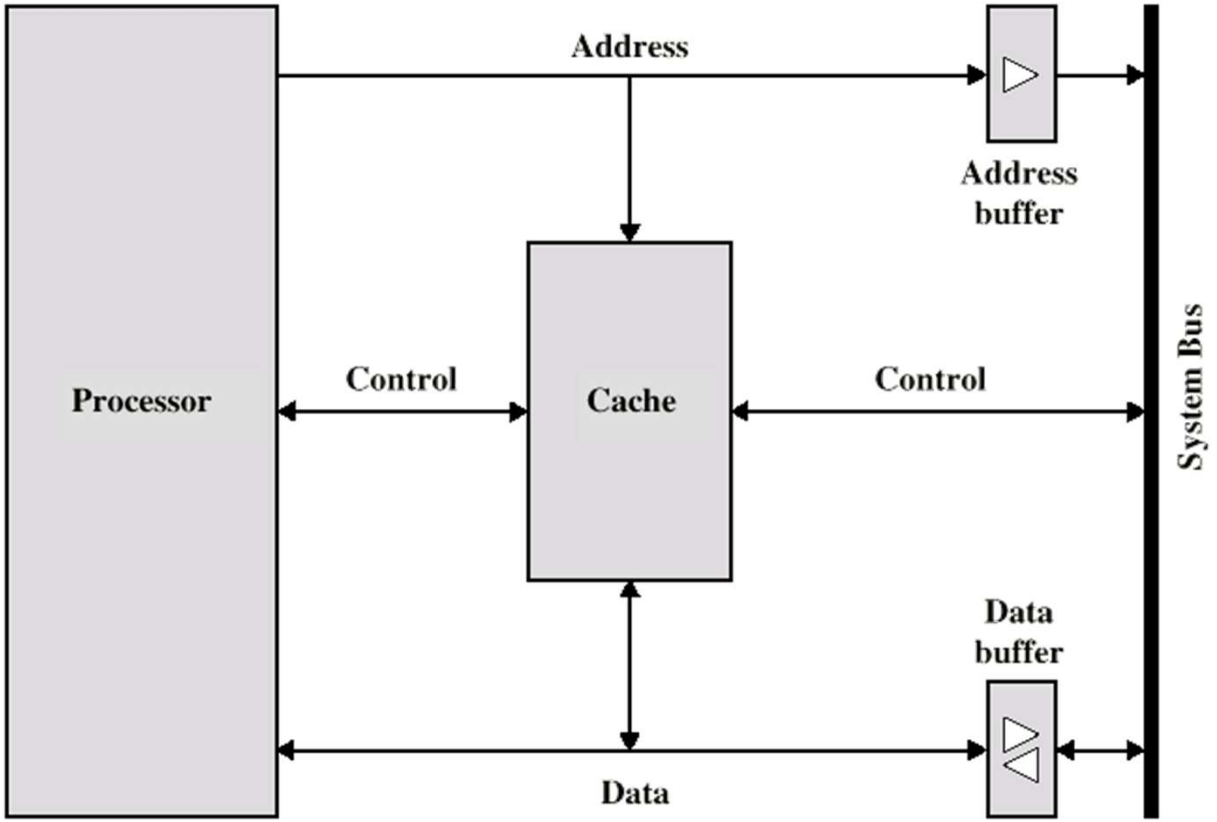


Operasi Cache

1) CPU menghasilkan alamat (Read Address/RA) dari word yang akan dibaca
2) Periksa apakah blok yang mengandung RA ada di cache
3) Jika Ya, ambil dari cache (fast), kembali
4) Jika Tidak, akses memori utama untuk mengambil blok yang dibutuhkan
5) Set cache untuk mengakses blok ini
6) Muat blok ke cache, dan bersiap untuk diakses CPU



Organisasi Cache Umumnya

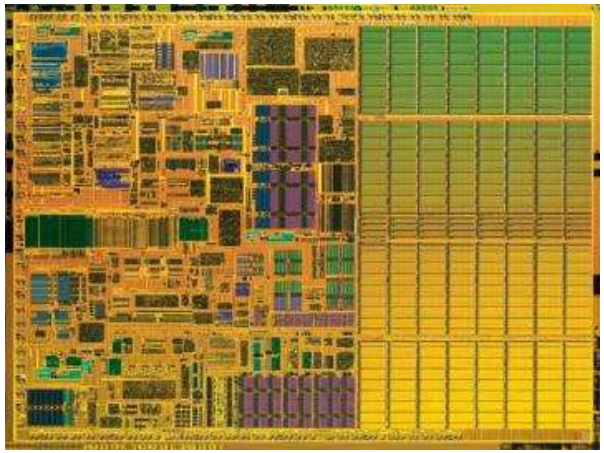


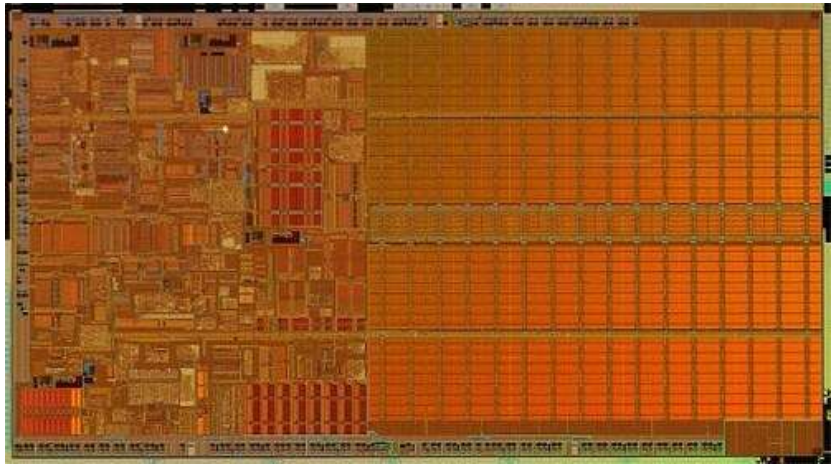
Elemen-elemen Perancangan Cache

- Ukuran Cache
- Fungsi Pemetaan
- Algoritma Replacement
- Kebijaksanaan Penulisan (Write Policy)
- Ukuran Baris Instruksi
- Jumlah dari Cache

Ukuran Cache

- Biaya
  - Semakin besar cache semakin mahal → bukti lihat daftar harga
- Kecepatan
  - Semakin besar cache semakin besar peluang CPU menemukan data di cache → akses CPU lebih cepat (lebih sedikit block swapping)
  - Mengakses data (proses pengkodean alamat) di cache memerlukan waktu → semakin besar cache akan menyebabkan waktu pencarian lebih lama





Centrino Class Processor

1. Banias 1MB L2 Cache
2. Dothan 2MB L2 Cache

Fungsi Pemetaan

- Pemetaan Langsung
- Pemetaan Asosiatif
- Pemetaan Set asosiatif

Misal :

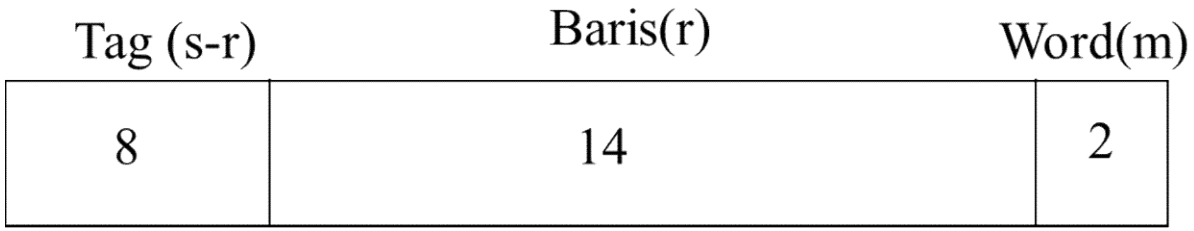
- Ukuran Cache 64KB, Ukuran blok 4 Byte, Baris intruksi di cache 16K (64KB/4B=16K), Ukuran memori utama 16MB
- Pengalamatan memori utama 24 bit
- Jumlah blok di memori utama 4M

Pemetaan Langsung (Direct Mapping)

- Setiap blok dari memori utama dipetakan ke hanya satu baris cache
  contoh : jika sebuah blok ada di cache, maka akan terletak di lokasi tertentu
- Bloks dari memori terhubungkan dengan baris cache
- Jumlah baris dapat dihitung dari alamat yang diberikan

Direct Mapping Address Structure

- Alamat 24 bit (s+w)
- Identifier word 2 bit (4 byte blok)
- Identifier blok 22 bit
  - Baris 14 bit (w)
  - Tag 8 bit (=22-14)
- 2 blok di baris yang sama mempunyai tag field yang berbeda
- Pemeriksaan isi cache dilakukan dengan melihat baris dan tag

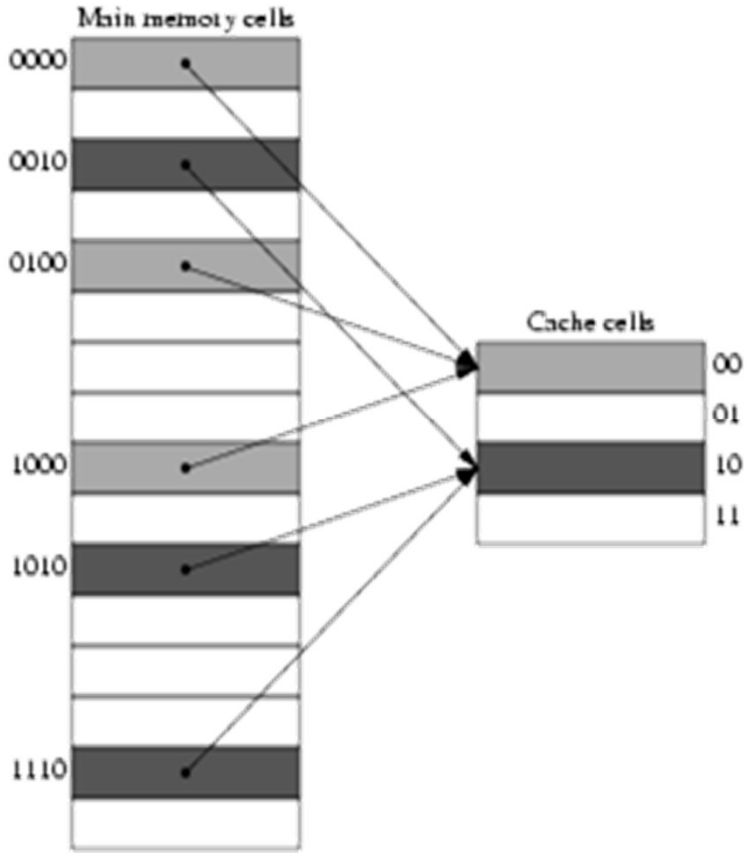


Tabel Baris Cache Pemetaan Langsung

Baris cache	Blok memori utama
0	0, m, 2m, 3m…2^s-m
1	1,m+1, 2m+1…2^s-m+1
…	
m-1	m-1, 2m-1,3m-1…2^s-1

m= 2^{14} (16K)

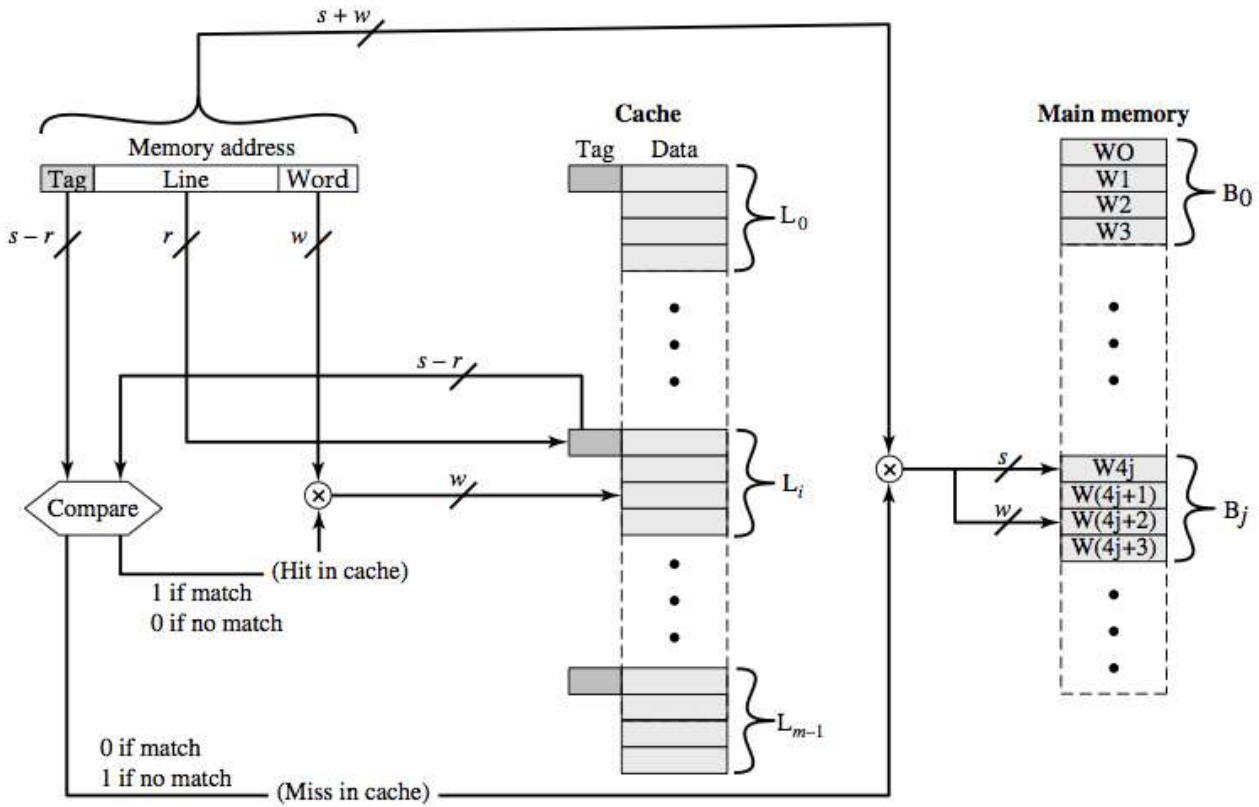
Contoh Pemetaan Langsung



+ - dari Pemetaan Langsung

- Sederhana
- Murah
- Lokasi tetap untuk setiap blok
  - Jika sebuah program mengakses 2 blok yang dipetakan ke baris yang sama berulang kali, maka cache miss akan sangat tinggi

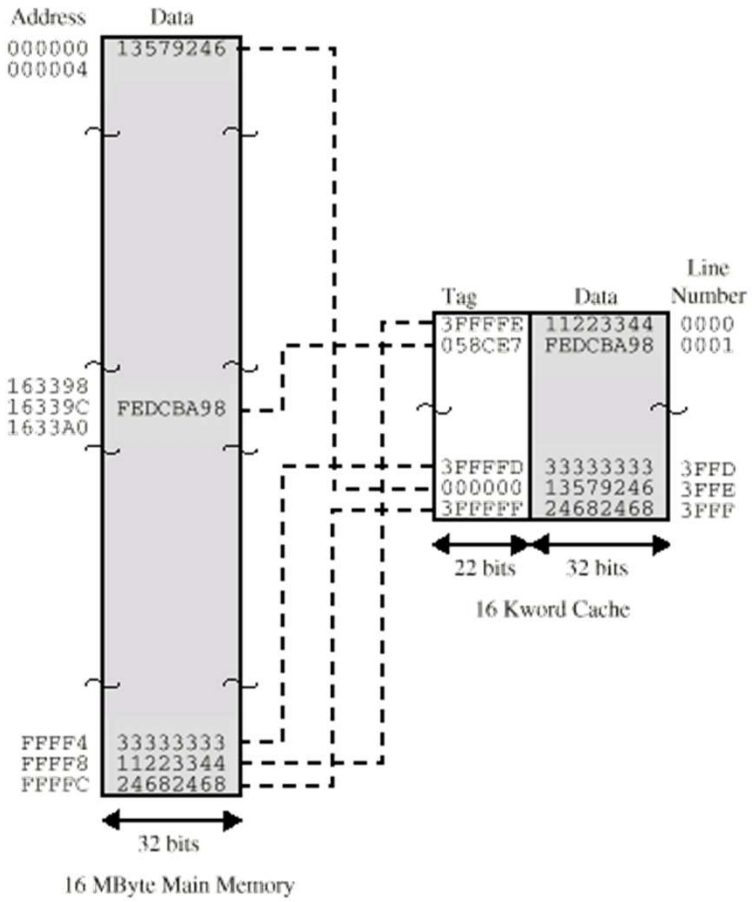
Direct Mapping



Pemetaan Asosiatif

- Sebuah blok dari memori utama dapat dimuat ke sembarang baris cache
- Alamat memori diterjemahkan sebagai tag dan word
- Tag merupakan identifikasi unik dari memori blok

Contoh Pemetaan Asosiatif

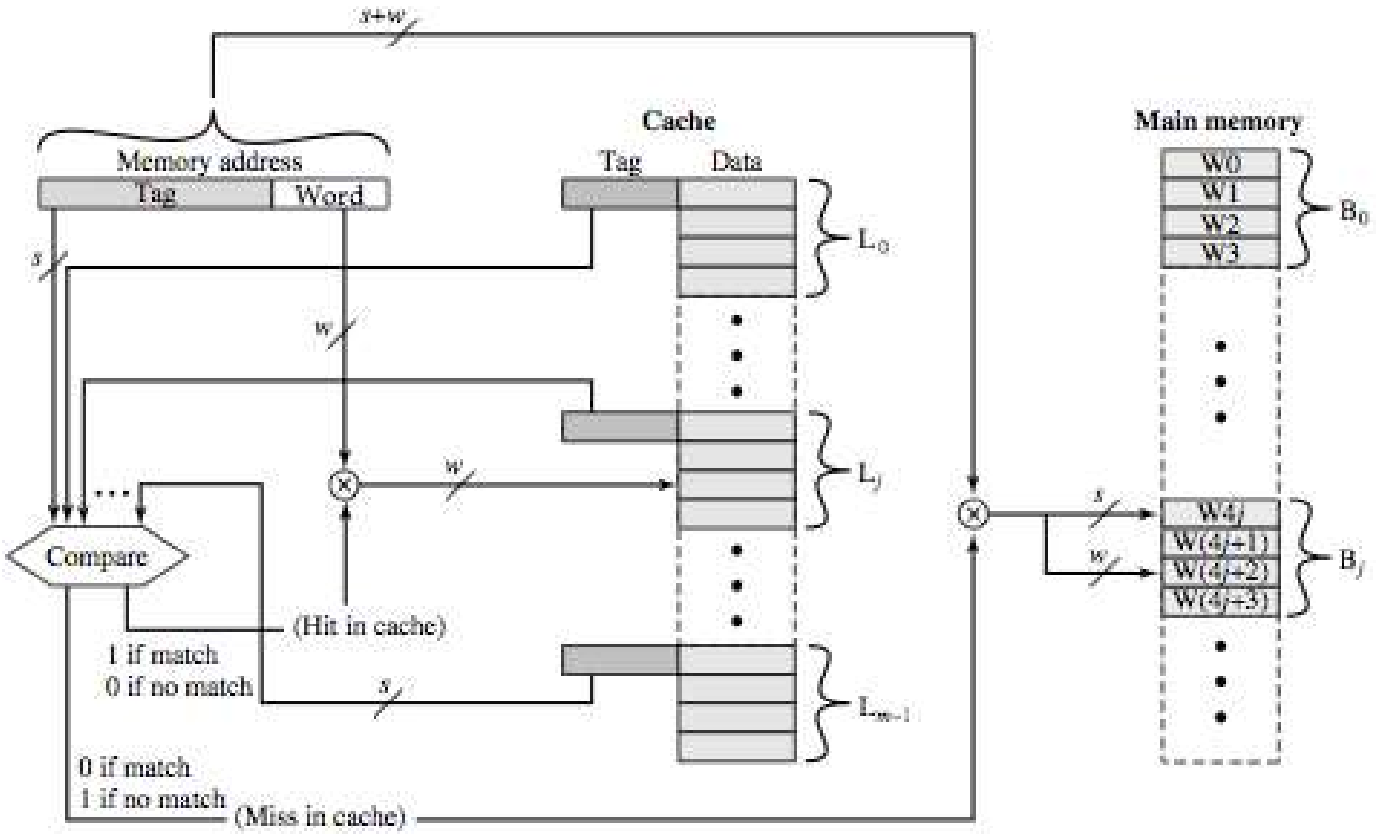


Struktur Pengalamatan Asosiatif

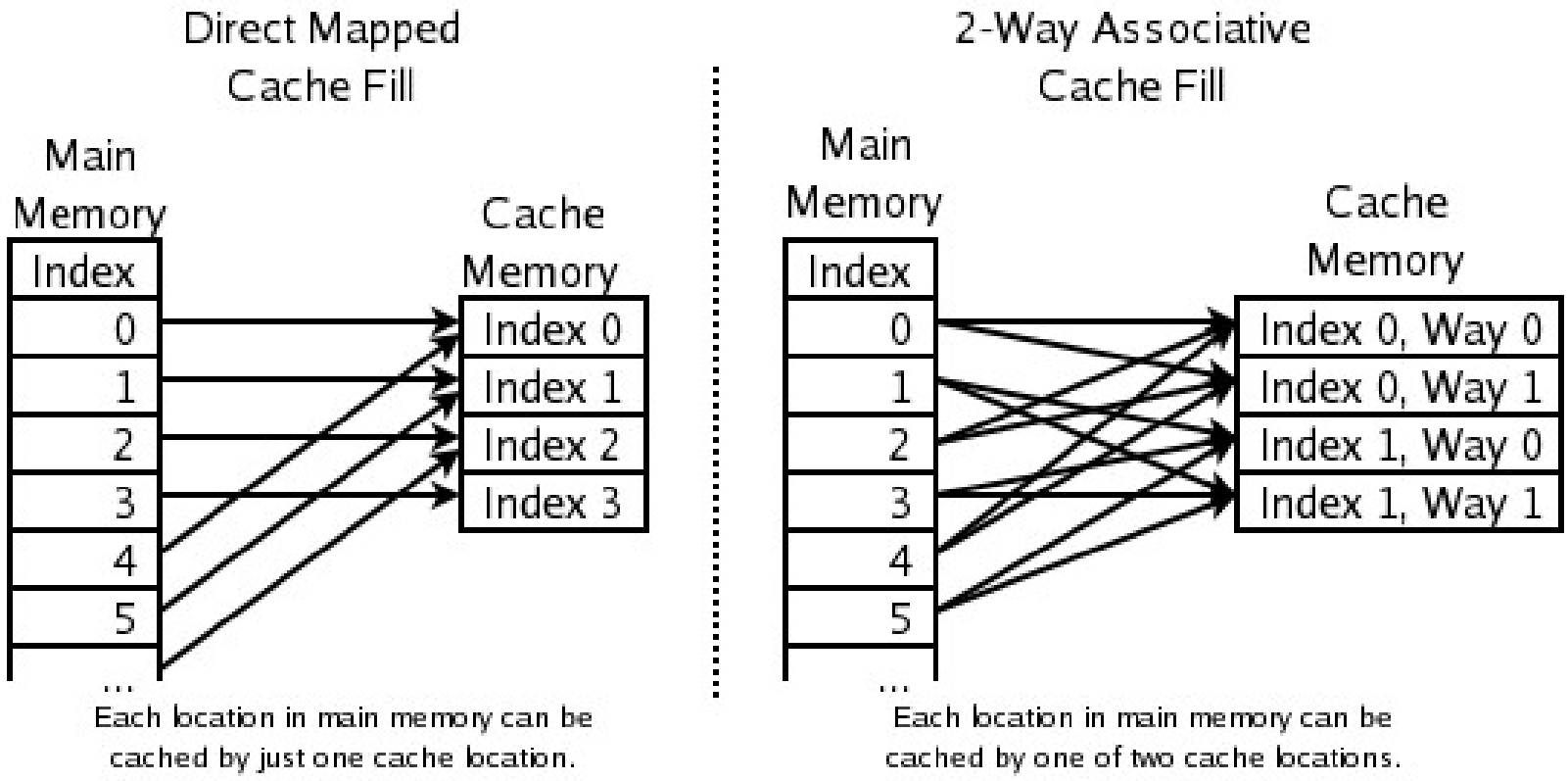
- Tag 22 bit tag disimpan dengan setiap blok data 32 bit
- Bandingkan field tag dengan input tag di cache untuk melihat apakah terjadi hit
- 2 bit alamat LSB mengidentifikasi word 16 bit mana yang diperlukan dari blok data 32 bit

Al.Cache	Tag	Data	Baris
16339C	058CE7	FEDCBA98	0001
FFFFFC	3FFFFF	24682468	3FFF

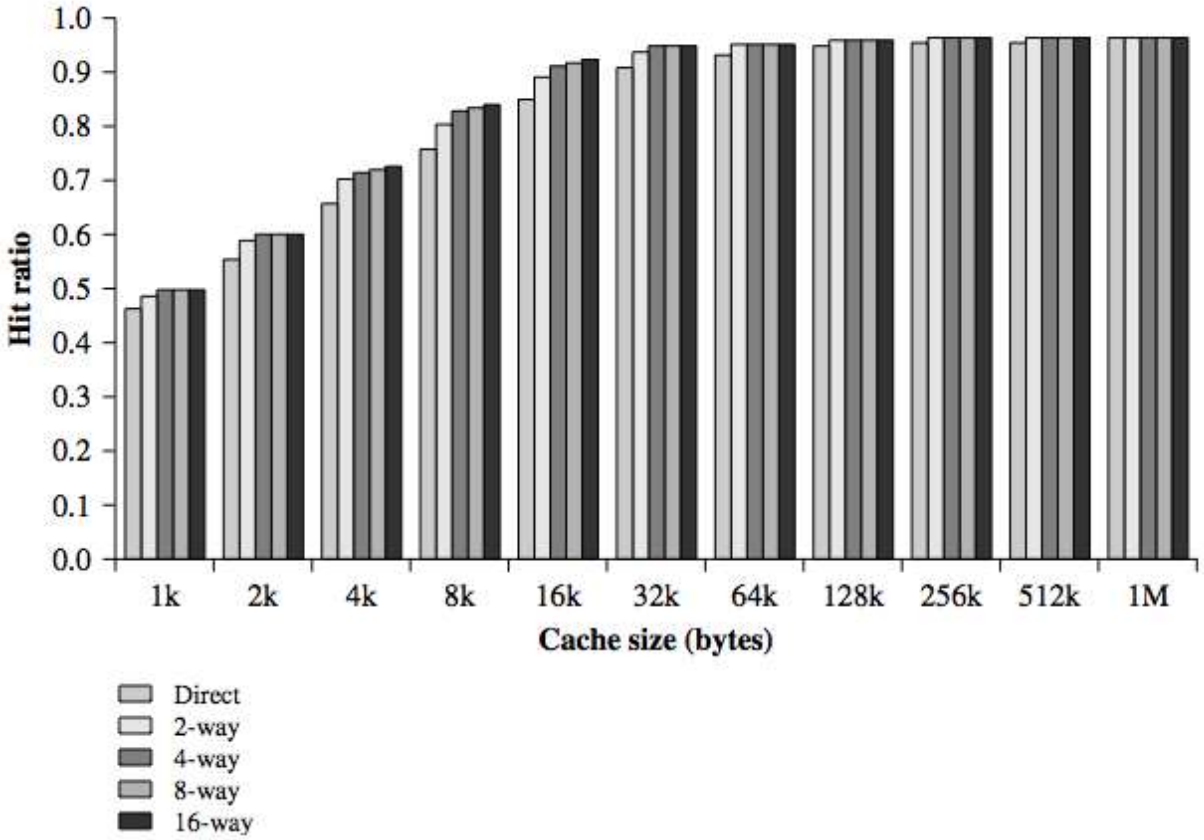
Pemetaan Asosiatif



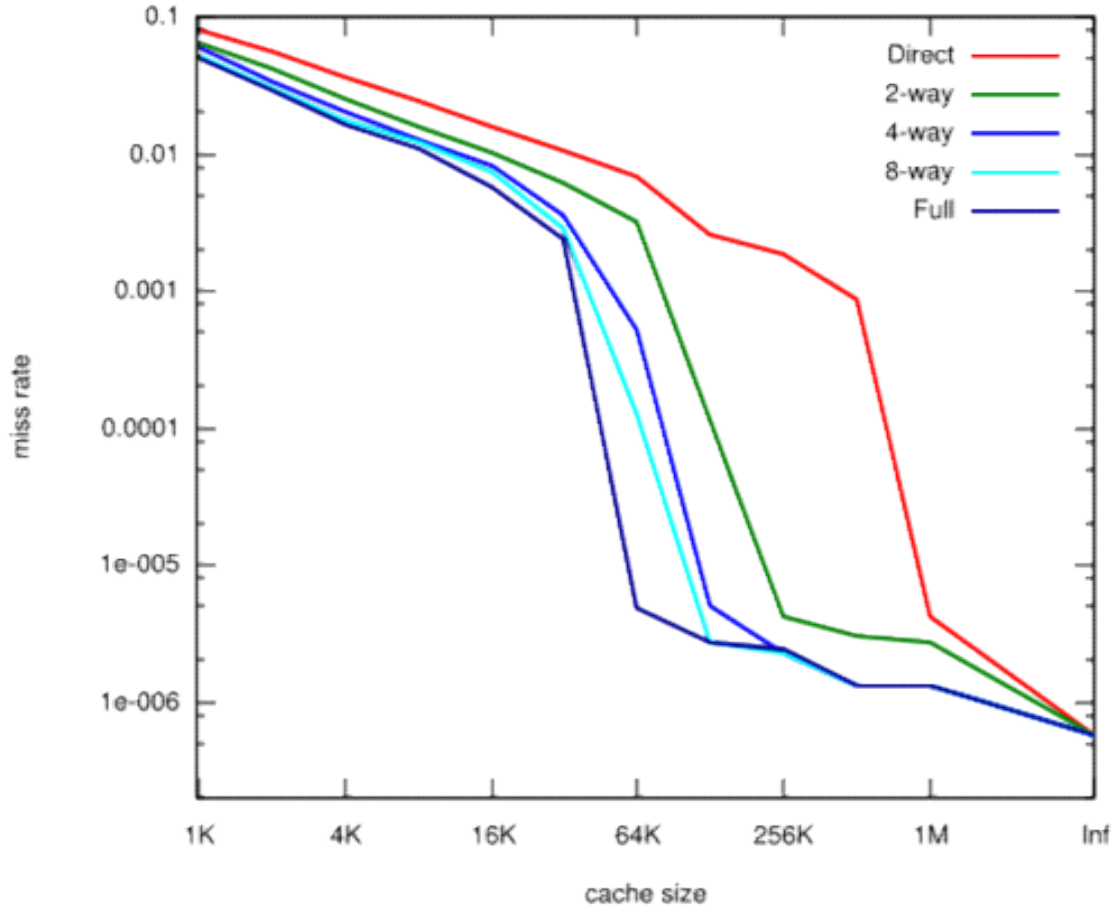
Pemetaan Set Asosiatif 2 Arah



Ratio (Hit Rate vs Cache Size)



Ratio (Miss rate VS cache size)



Algoritma Replacement (1)
Pemetaan Langsung

- Tidak ada pilihan
- Setiap blok terpetakan ke satu baris
- Ganti baris tersebut

Algoritma Replacement (2)
Asosiatif & Set Asosiatif

- Dibuat di h/w agar lebih cepat
- First in first out (FIFO)
  - Mengganti blok yang terlama ada di cache
- Least frequently used
  - Mengganti blok yang mendapat hit paling sedikit
- Random

Kebijakan (Policy) Penulisan

- Data di cache dan data di memori utama harus terkini (up to date)
- Banyak divais dapat mengakses memori utama (I/O, dan CPU)
- Sistem dengan banyak CPU dapat hak akses cache individu
- Jika sebuah word di ubah disuatu lokasi, maka word di lokasi lain harus di update

Write through

- Semua penulisan harus langsung ke memori utama selain ke cache
- + : sederhana
- - :
  - Trafik banyak
  - Penulisan lebih lambat

Analog dengan optimized for quick removal di USB

Write back

- Update permulaan hanya dilakukan di cache
- Saat update terjadi bit update di set
- Jika sebuah blok akan diganti, penulisan ke memori utama akan dilakukan untuk blok yang bit updatenya sudah di set
- +: penulisan ke memori utama minimal
- -:
  - Sirkit menjadi lebih komplek dan menyebabkan bottleneck
  - Data di memori utama menjadi tidak valid, sehingga I/O harus mengakses data di cache
  - Isi cache lain menjadi tidak sinkron
- *Analog dengan optimized for performance di USB*

Perbandingan Performa

- Misal akses RAM 100ns, cache 10ns terdapat 4 baris cache, ada 4 kali operasi penulisan
- Jika menggunakan writetrough (data ditulis ke cache dan ram) maka waktu yang dibutuhkan adalah 4 penulisan x 4 baris x 100 ns = 1600 ns
- Jika menggunakan writeback (data hanya ditulis ke cache) maka waktu yang dibutuhkan adalah 4 x 4 baris x 10 ns = 160 ns + 400 ns (penulisan final)

Ukuran Baris

- Ukuran blok ↑, rasio hit ↑
- Ukuran blok ↑, jumlah blok di cache ↓
- Ukuran blok ↑, relevansi word ↓

Jumlah Cache

- Cache Multilevel
  - L1: on-chip, L2: on-chip, L3: external cache
  - Tidak ada akses bus sistem antara CPU dan L1,L2,L3
- Cache Tunggal vs. Terpisah
  - Tunggal: rate hit tinggi, mudah di implementasikan
  - Terpisah: satu cache untuk instruksi, satu untuk data data

Apakah betul cache akan meningkatkan performa sistem ?

- Asumsi :
  - Akses memori (RAM) = 100 ns
  - Akses cache = 10 ns (waktu pemetaan + waktu pencarian + waktu reaksi memori cache + faktor X)
  - Cache hit = 99%
- Maka kecepatan akses suatu data adalah :

  (0,99*10ns) + (0,01*(10ns+100ns)) = 9,9ns + 1,1 = **11 ns**
- Kesimpulan : dengan cache rata-rata waktu akses sistem akan turun (contoh : dari 100 ns → 11 ns)

Cache lebih dari 1 Level

Asumsi :

- Akses memori (RAM) = 100 ns
- Akses cache = 10 ns (waktu pemetaan + waktu pencarian + waktu reaksi memori cache + faktor X)
- Cache hit = 99%

Maka waktu rata-rata :

- Data ada di LI = 0,99 x 10 ns = 9,9 ns
- Data ada di L2 = 0,01 x 0,99 x (10+10)ns = 0,198ns
- Data ada di RAM = 0,01 x 0,01 x (10+10+100)ns = 0,012 ns

→ Waktu rata-rata akses = 9,9 + 0,198+0,012 = **10,11** ns

Pengaruh Banyaknya Cache & Cache Hit Ratio

Cache	0.9	0.99
L1	20	11
L2	12	10.11
L3	11.2	10,1011

Multicore Organization

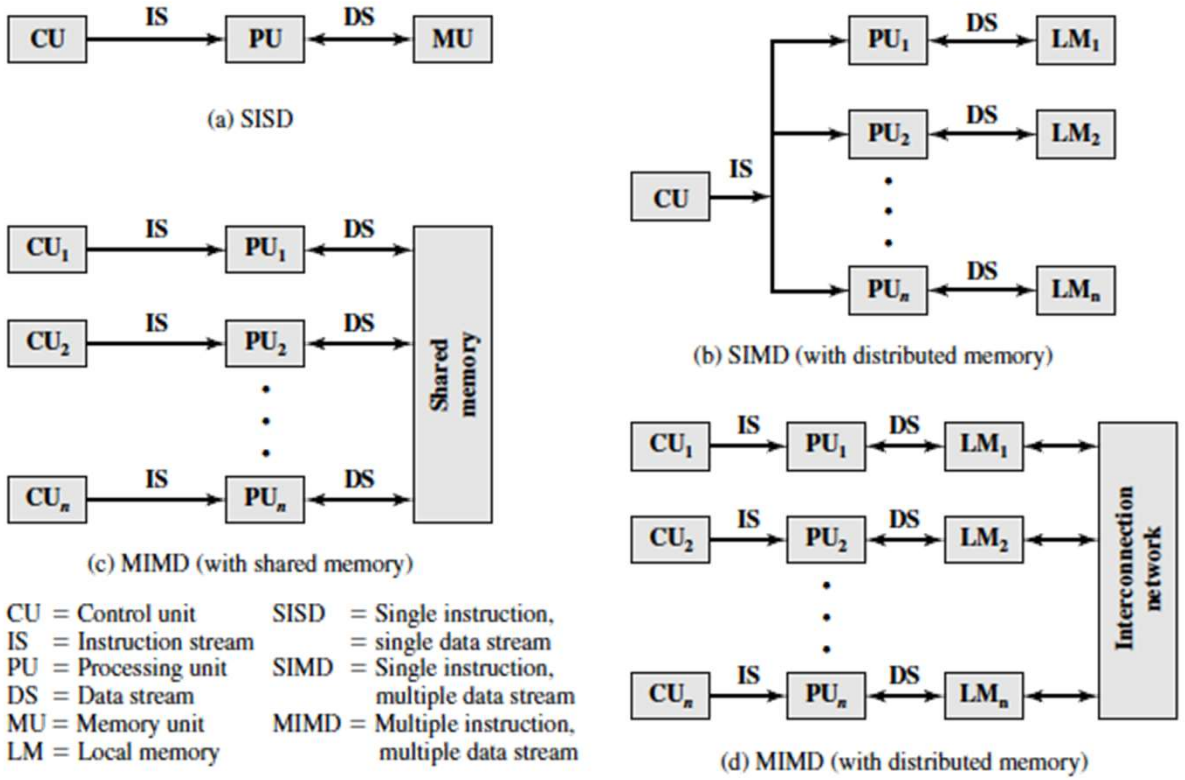


Figure 17.2 Alternative Computer Organizations

Shared Memory Multiprocessor

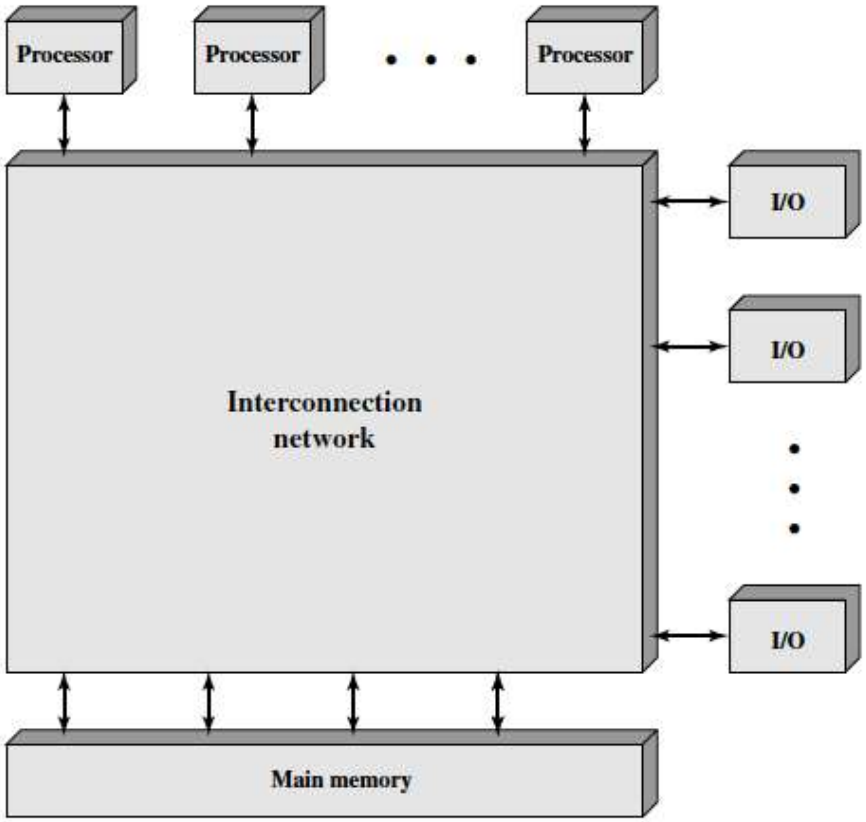


Figure 17.4 Generic Block Diagram of a Tightly Coupled Multiprocessor

Symmetric Multiprocessor

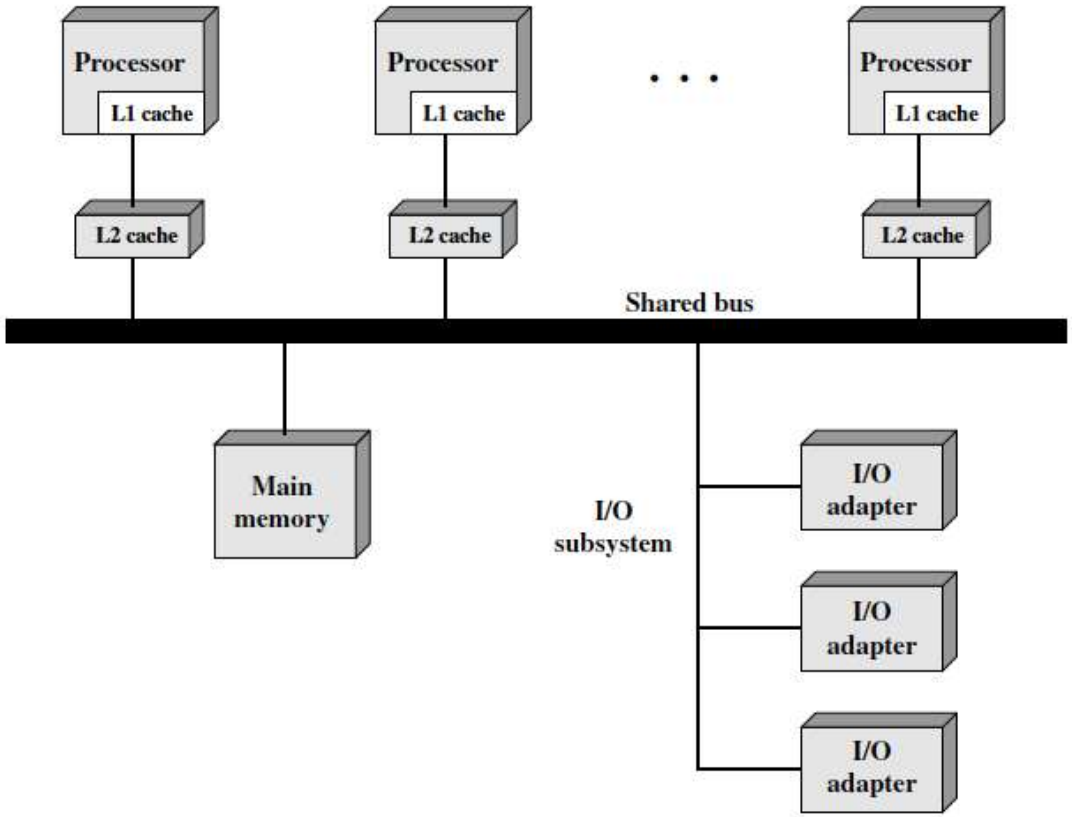


Figure 17.5 Symmetric Multiprocessor Organization