CG 2007 Microprocessor Systems Lecture 11

Pipelining and Memory Hierarchy

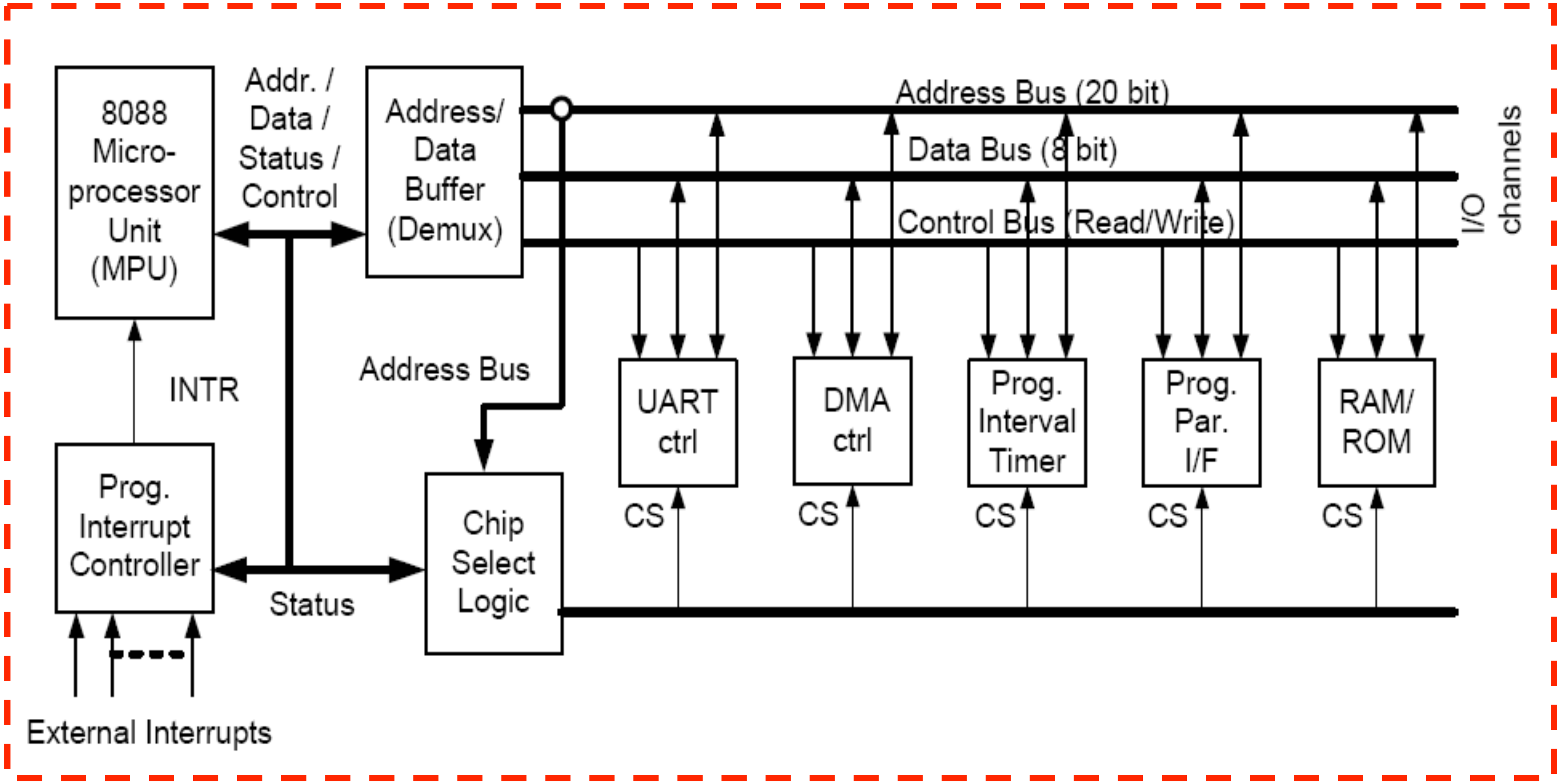
Dr. Ha Yajun
(E1-08-13, *elehy@nus.edu.sg*)

Lecture 11: Objectives and Outline

- **Objectives**
  - Understand how the pipelining, memory hierarchy concepts can be used to improve the microprocessor system performance

- **Outline**
  - Pipelining Concepts
  - Memory Hierarchy Concepts

Simplified System Architecture of an IBM Compatible PC



Outline: Pipelining and Memory Hierarchy

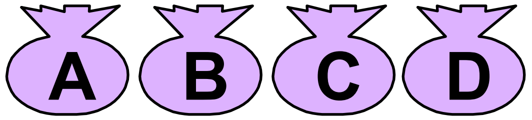
– Pipelining Concepts

– Memory Hierarchy Concepts

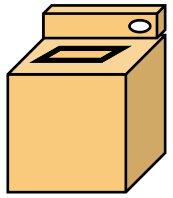
Pipelining is Natural!

Laundry Example:

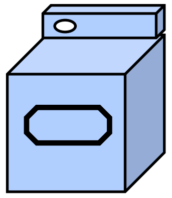
- **Ann, Brian, Cathy, Dave each has one load of clothes to wash, dry, and fold**



- **Washer takes 30 minutes**

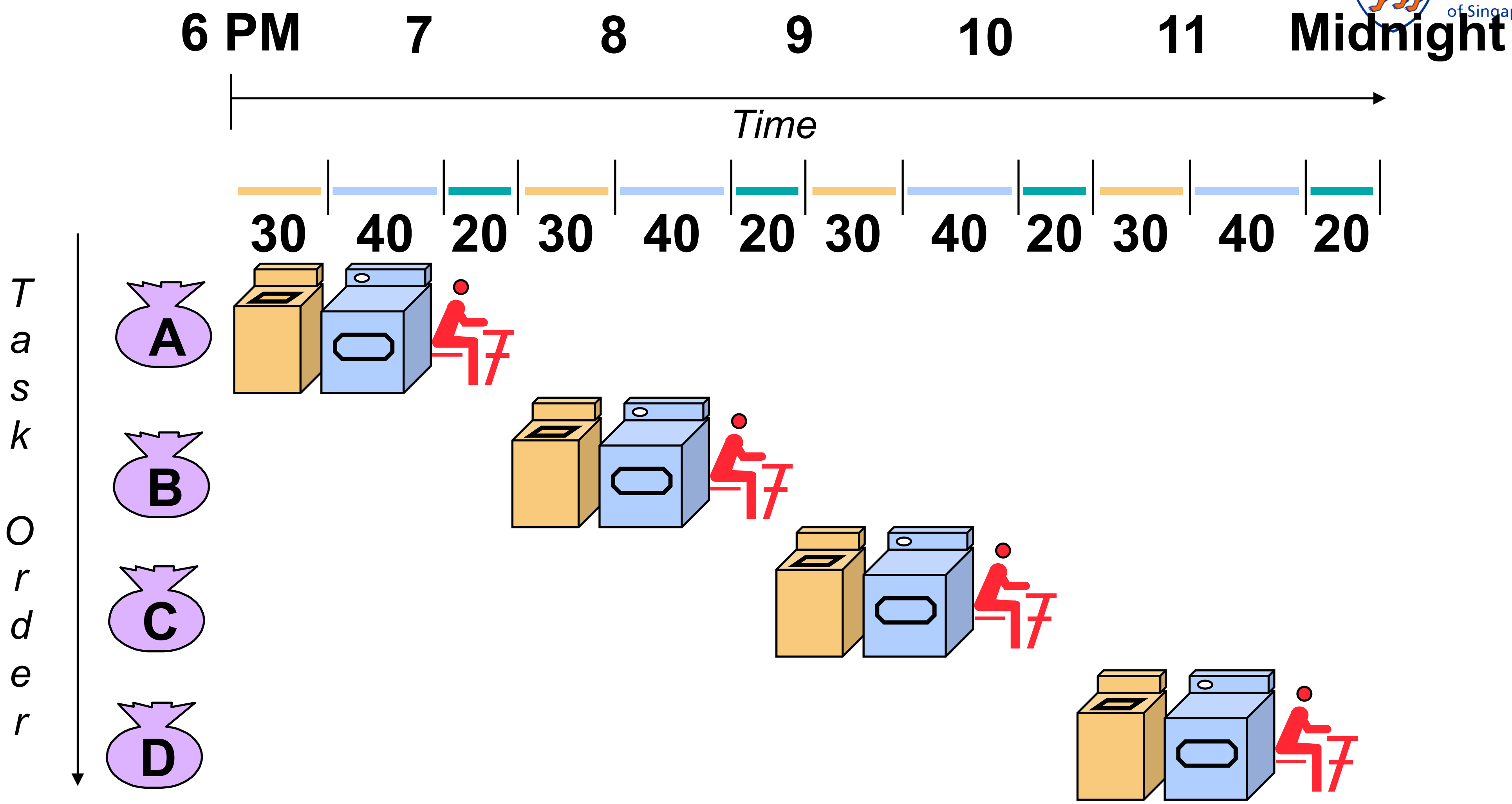


- **Dryer takes 40 minutes**



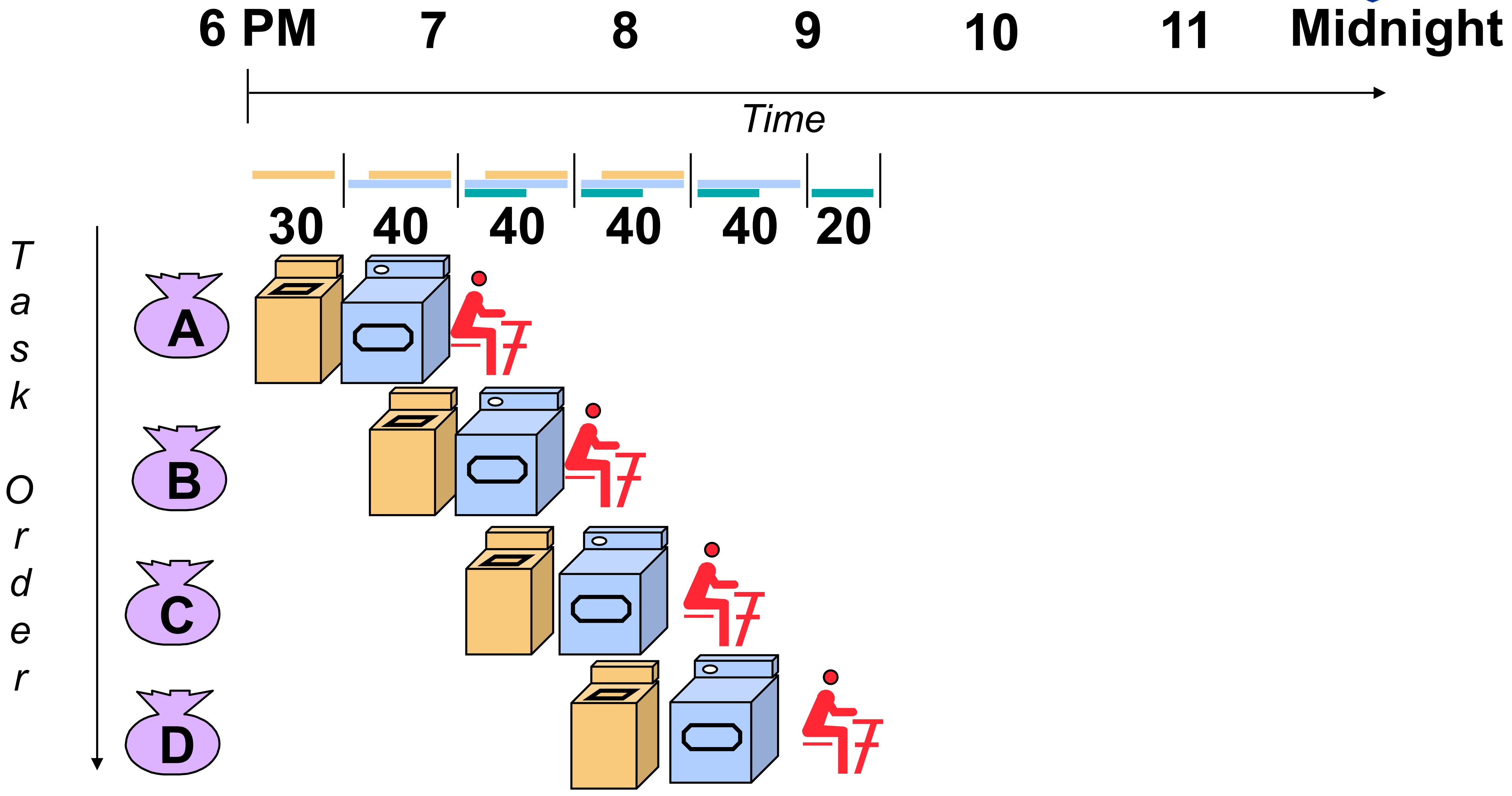
- **"Folder" takes 20 minutes**

Sequential Laundry



- **Sequential laundry takes 6 hours for 4 loads**

Pipelined Laundry: Start work ASAP



•**Pipelined laundry takes 3.5 hours for 4 loads**

Latency and Throughput in Pipelining

- **Latency vs. Throughput**
  - Latency is the time from the start to the end to finish ONE load
  - Throughput is the number of loads finished per unit time

- **Questions**
  - What is the latency for 1 load in both cases?

    Latency using sequential approach = 90 min

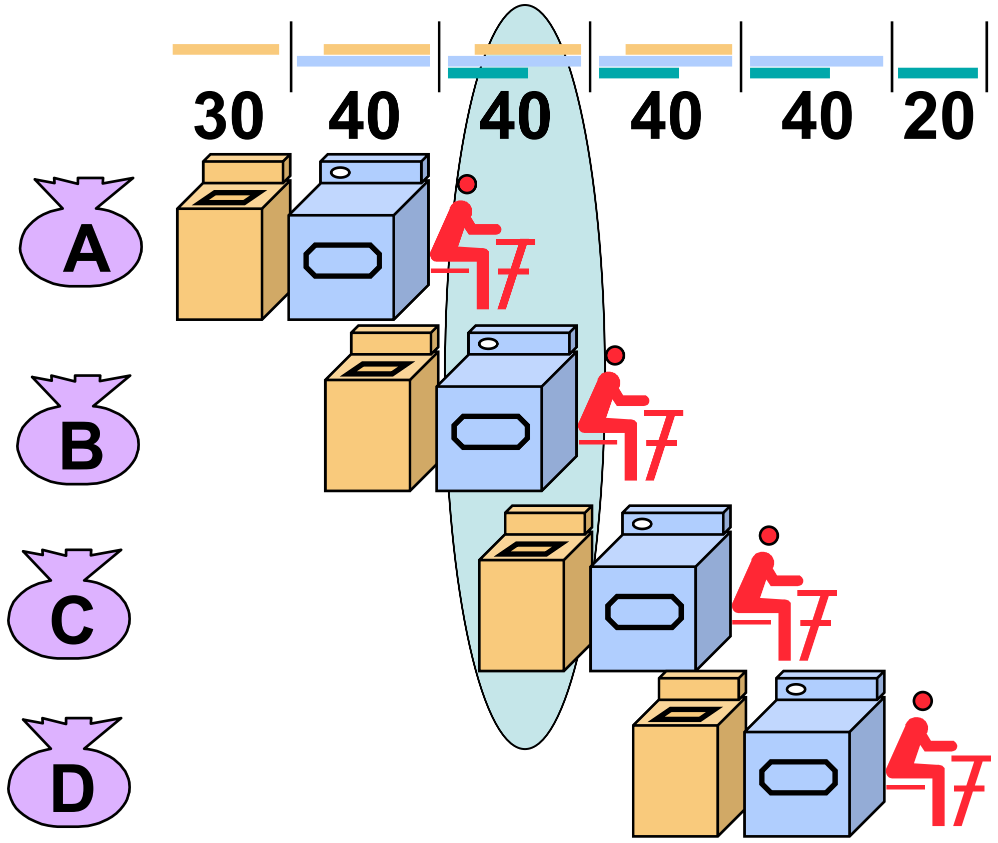
    Latency using pipelining approach ~ 90 min
  - What is the throughput for 4 loads in both cases (Sequential and pipelining)?

    Throughput using sequential approach = 4 loads/6 hours = ~ 0.67 loads/hr

    Throughput using pipelining approach = 4 loads/3.5 hours = ~ 1.43 loads/hr

Pipelining doesn't help **latency** of a single task, but it helps **throughput** of entire workload!

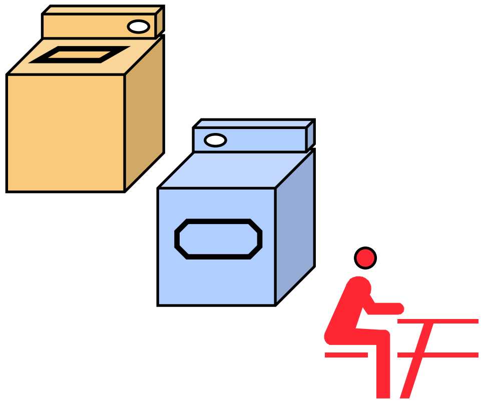
Resource Usage in Pipelining



Multiple tasks operate **simultaneously** using different resources

Pipelining Speedup

$$\text{Speedup} = \frac{\text{The time to finish tasks without using pipelining}}{\text{The time to finish tasks using pipelining}}$$

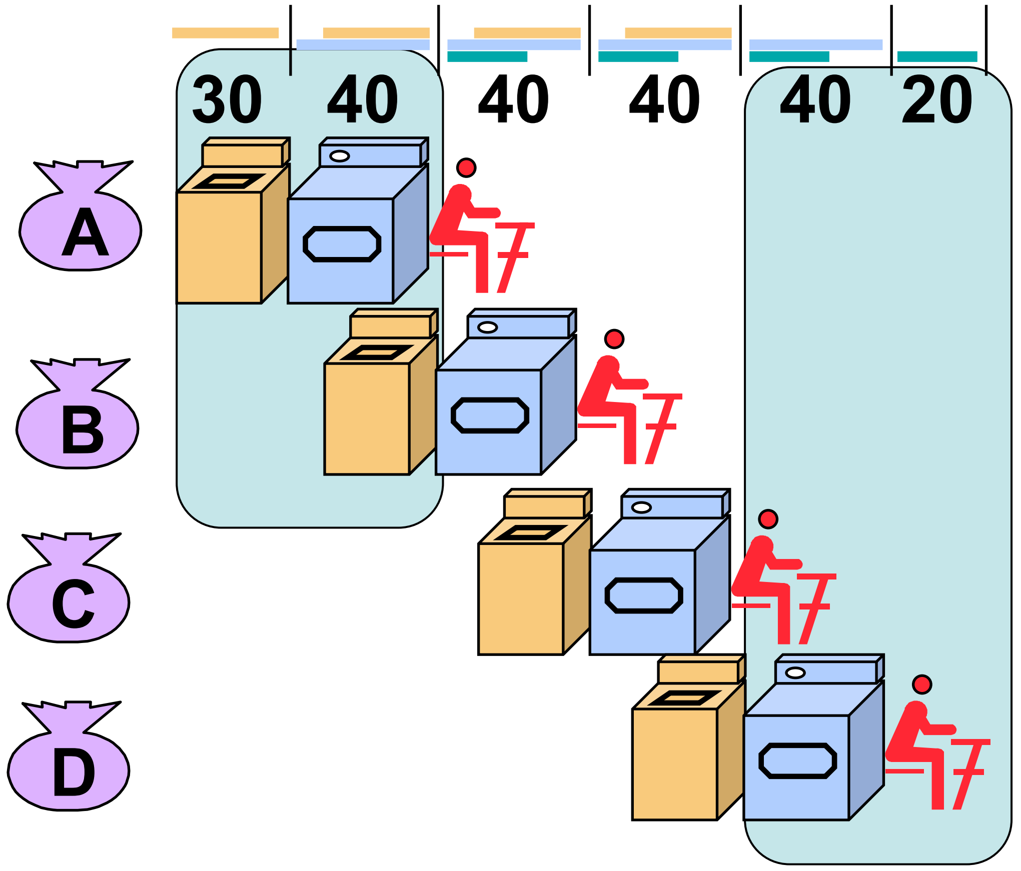


- Washer takes 30 minutes
- Dryer takes 40 minutes
- "Folder" takes 20 minutes

- Question
  - Will it affect the speedup if "Folder" also takes 40 minutes?

Unbalanced lengths of pipe stages reduces speedup!

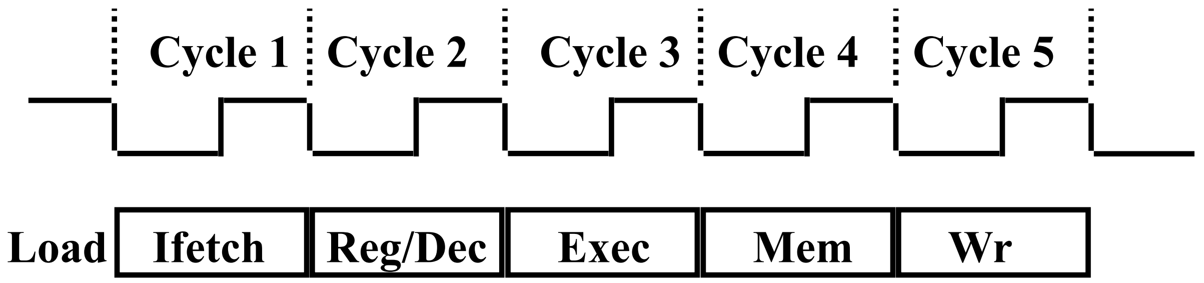
Filling and Draining Pipelines Reduce Speedup



Time to "fill" pipeline reduces speedup

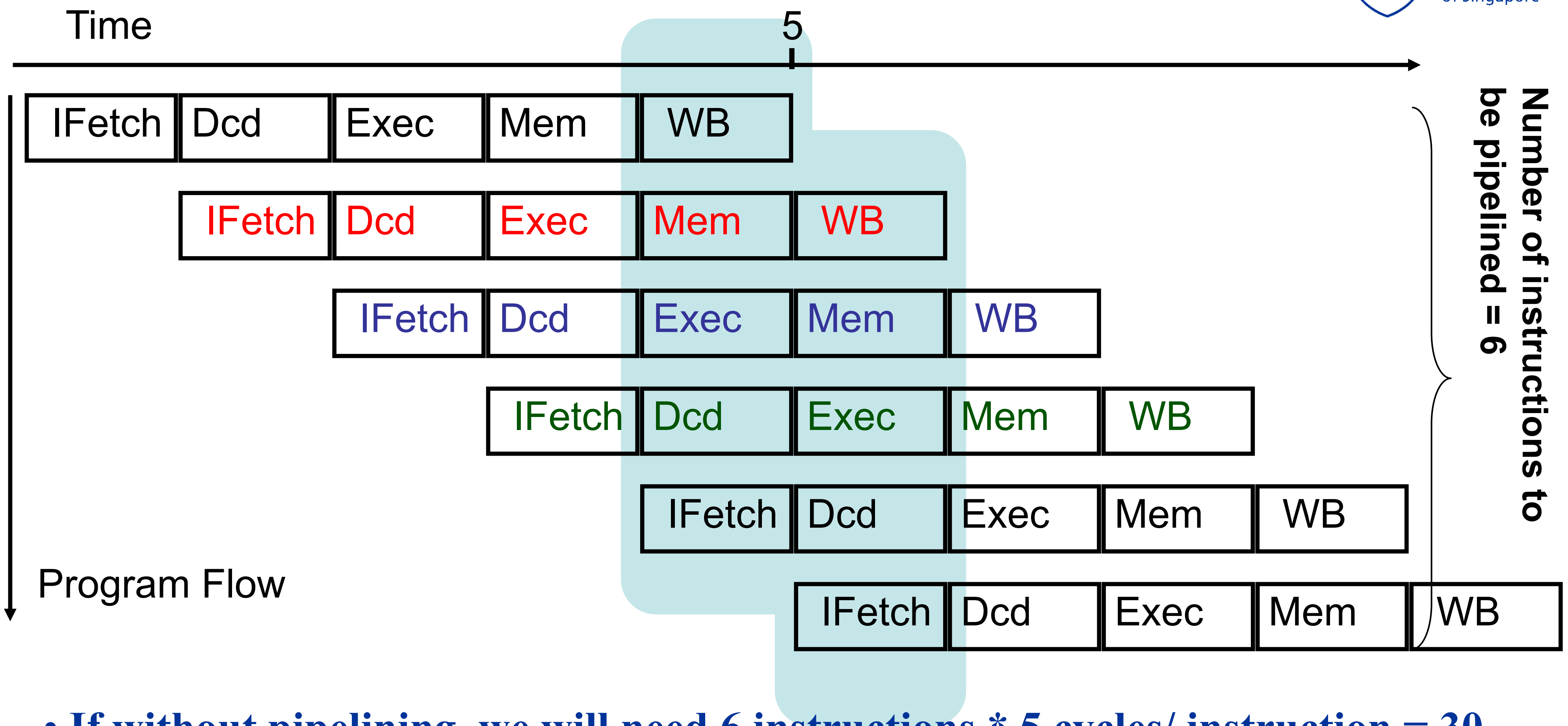
Time to "drain" it also reduces speedup

Pipelining an Instruction with Five Stages



- **Ifetch: Instruction Fetch**
  - Fetch the instruction from the Instruction Memory
- **Reg/Dec: Registers Fetch and Instruction Decode**
- **Exec: Execute the operation or calculate the memory address**
- **Mem: Read the data from the Data Memory**
- **Wr: Write the data back to the register file**

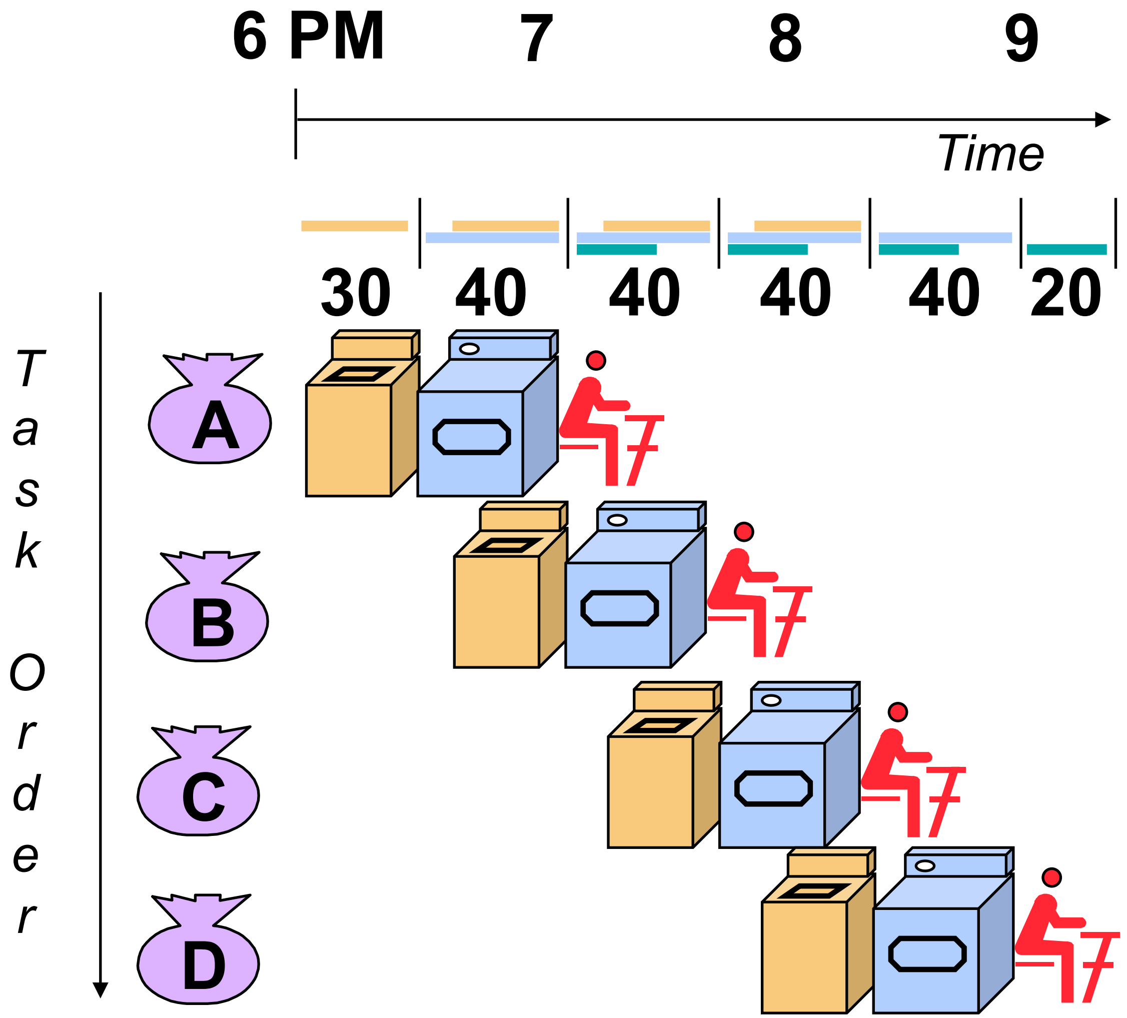
Conventional Pipelined Execution Representation



- If without pipelining, we will need 6 instructions * 5 cycles/ instruction = 30 cycles to finish the 6 instructions.
- With the pipelining, we just need 10 cycles to finish the 6 instructions.

Assumption: all the 5 pipe stages for an instruction execution are balanced.

Summary of Pipelining Lessons



- **Pipelining doesn’t help** latency **of single task, it helps** throughput **of entire workload**
- **Multiple tasks operating simultaneously using different resources**
- **Pipeline rate limited by slowest pipeline stage**
- **Unbalanced lengths of pipe stages reduces speedup**
- **Time to “fill” pipeline and time to “drain” it reduces speedup**

Outline: Pipelining and Memory Hierarchy

- Pipelining Concepts
- Memory Hierarchy Concepts
  - **Storage Devices and CPU-Memory Gap**
  - **Locality, Cache and Memory Hierarchy**
  - **Four Cache and Memory Hierarchy Questions**
  - **Practical Memory Hierarchy**

Storage Types

- **RAM (Random Access Memory)**
  - SRAM.
  - DRAM
- **ROM (Read Only Memory)**
  - PROM
  - EPROM
  - EEPROM.
- **Disk**

These are the three types of storage devices that can store your instructions and data! There have different features!

Random-Access Memory (RAM)

- **Key features**
  - RAM is packaged as a chip.
  - Basic storage unit is a cell (one bit per cell).
  - Multiple RAM chips form a memory module.
- **Static RAM (SRAM)**
  - Each cell stores 1 bit with a six-transistor circuit.
  - Retains value indefinitely, as long as it is kept powered.
  - Relatively insensitive to disturbances such as electrical noise.
  - Faster and more expensive than DRAM.
- **Dynamic RAM (DRAM)**
  - Each cell stores 1 bit with a capacitor and a transistor.
  - Value must be refreshed every 10-100 ms.
  - Sensitive to disturbances.
  - Slower and cheaper than SRAM.

SRAM vs. DRAM Summary

	Tran. per bit	Access time	Persist?	Sensitive?	Cost	Applications
SRAM	6	1X	Yes	No	100x	cache memories
DRAM	1	10X	No	Yes	1X	Main memories, frame buffers

Nonvolatile Memories

- **DRAM and SRAM are volatile memories**
  - Lose information if powered off.
- **Nonvolatile memories retain value even if powered off.**
  - Generic name is read-only memory (ROM).
  - Misleading because some ROMs can be read and modified.
- **Types of ROMs**
  - Programmable ROM (PROM)
  - Erasable programmable ROM (EPROM)
  - Electrically erasable PROM (EEPROM)
  - Flash memory
- **Firmware**
  - Program stored in a ROM

    Boot time code, BIOS (basic input/output system)

Storage Trends

SRAM

metric	1980	1985	1990	1995	2000	2004
$/MB	19,200	2,900	320	256	100	*4*
access (ns)	300	150	35	15	2	*0.5*

DRAM

metric	1980	1985	1990	1995	2000	*2004*
$/MB	8,000	880	100	30	1	*0.1*
access (ns)	375	200	100	70	60	*50*
typical size(MB)	0.064	0.256	4	16	64	*512*

Disk

metric	1980	1985	1990	1995	2000	*2004*
$/MB	500	100	8	0.30	0.05	*0.00005*
access (ms)	87	75	28	10	8	*5*
typical size(MB)	1	10	160	1,000	9,000	*80,000*

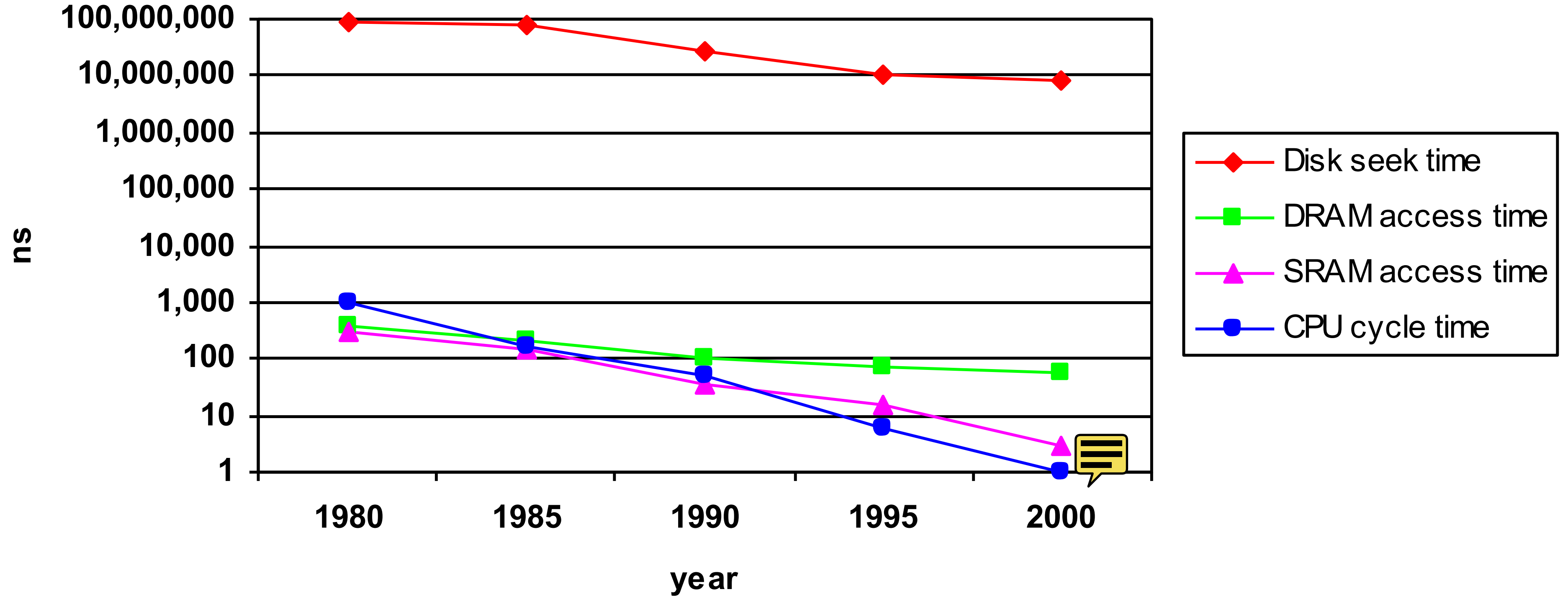
CPU Clock Rates

	1980	1985	1990	1995	2000	*2000:1980*
processor	8080	286	386	Pent	P-III	
clock rate(MHz)	1	6	20	150	750	750
cycle time(ns)	1,000	166	50	6	1.6	750

Dell Optiplex GX755 Configuration (sell at NUS CO-OP in 2008):

Intel Core 2 Duo Processor 2.33GHZ
4MB L2 Cache
250GB 7,200rpm hard disk
4GB DDR2 RAM

The CPU-Memory Gap



The increasing gap between DRAM, disk, and CPU speeds!

Outline: Pipelining and Memory Hierarchy

- Pipelining Concepts
- Memory Hierarchy Concepts
  - **Storage Devices and CPU-Memory Gap**
  - **Locality, Cache and Memory Hierarchy**
  - **Four Cache and Memory Hierarchy Questions**
  - **Practical Memory Hierarchy**

Locality

- **Principle of Locality:**
  - Programs tend to reuse data and instructions near those they have used recently, or that were recently referenced themselves.
  - Temporal locality: Recently referenced items are likely to be referenced in the near future.
  - Spatial locality: Items with nearby addresses tend to be referenced close together in time.

```
       MOV SI, 0200H
       MOV CX, 0010H
LOOP:  ADD AX, [SI]
       INC SI
       DEC CX
       JNZ LOOP
```

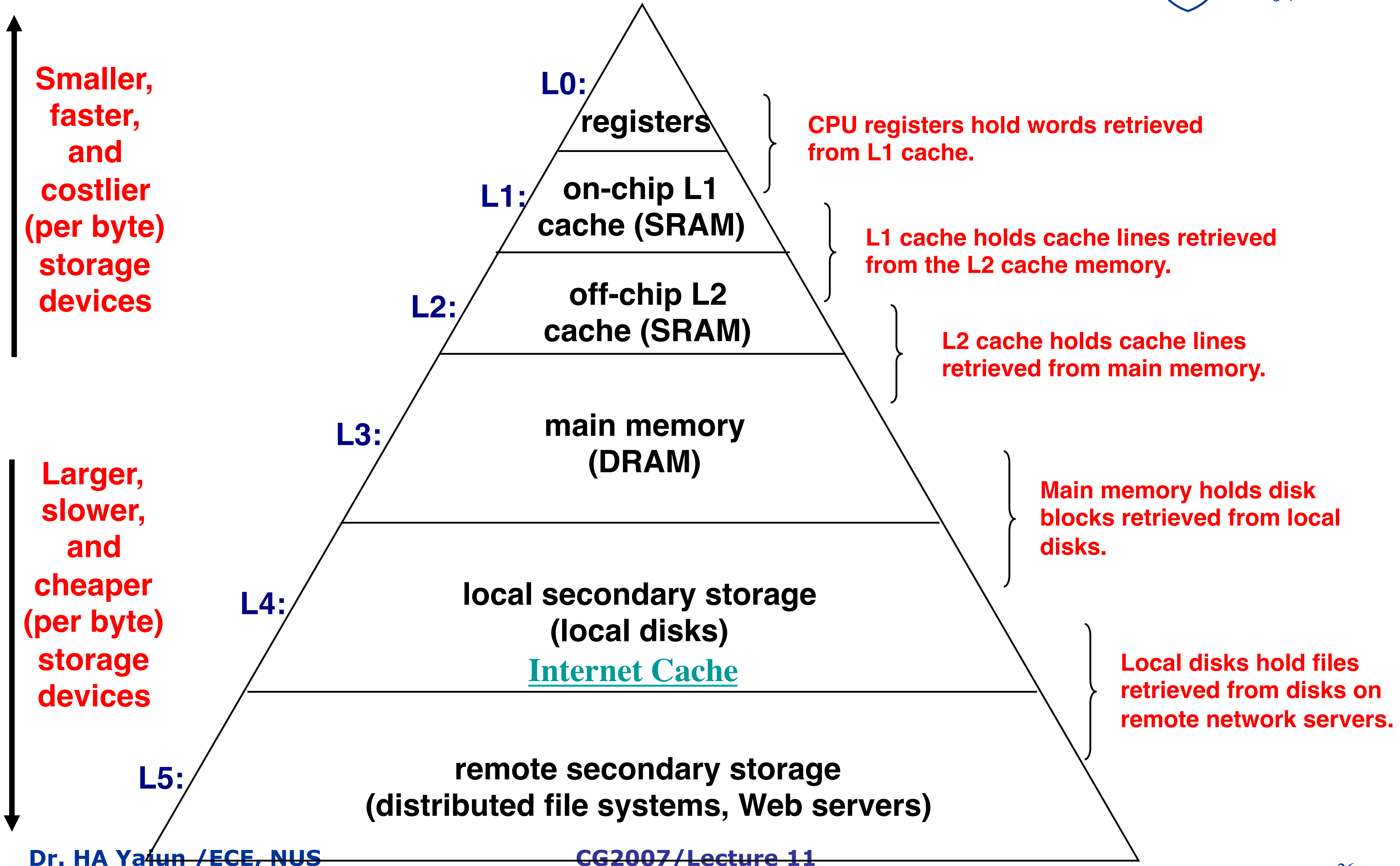
Locality Shown in the Example Code:

- **Data**
  - A group of data pointed by SI is read: **Spatial locality**
- **Instructions**
  - 4 continuous instructions in the loop will be executed repeatedly: **Spatial locality**
  - 4 instructions in the loop will be re-executed shortly: **Temporal locality**

Memory Hierarchies

- **Some fundamental and enduring properties of hardware and software:**
  - Fast storage technologies cost more per byte and have less capacity.
  - The gap between CPU and main memory speed is widening.
  - Well-written programs tend to exhibit good locality.
- **These fundamental properties complement each other beautifully.**
- **They suggest an approach for organizing memory and storage systems known as a memory hierarchy.**

An Example Memory Hierarchy



Smaller, faster, and costlier (per byte) storage devices
Larger, slower, and cheaper (per byte) storage devices
L0:
registers
L1:
on-chip L1 cache (SRAM)
L2:
off-chip L2 cache (SRAM)
L3:
main memory (DRAM)
L4:
local secondary storage (local disks)
Internet Cache
L5:
remote secondary storage (distributed file systems, Web servers)
CPU registers hold words retrieved from L1 cache.
L1 cache holds cache lines retrieved from the L2 cache memory.
L2 cache holds cache lines retrieved from main memory.
Main memory holds disk blocks retrieved from local disks.
Local disks hold files retrieved from disks on remote network servers.

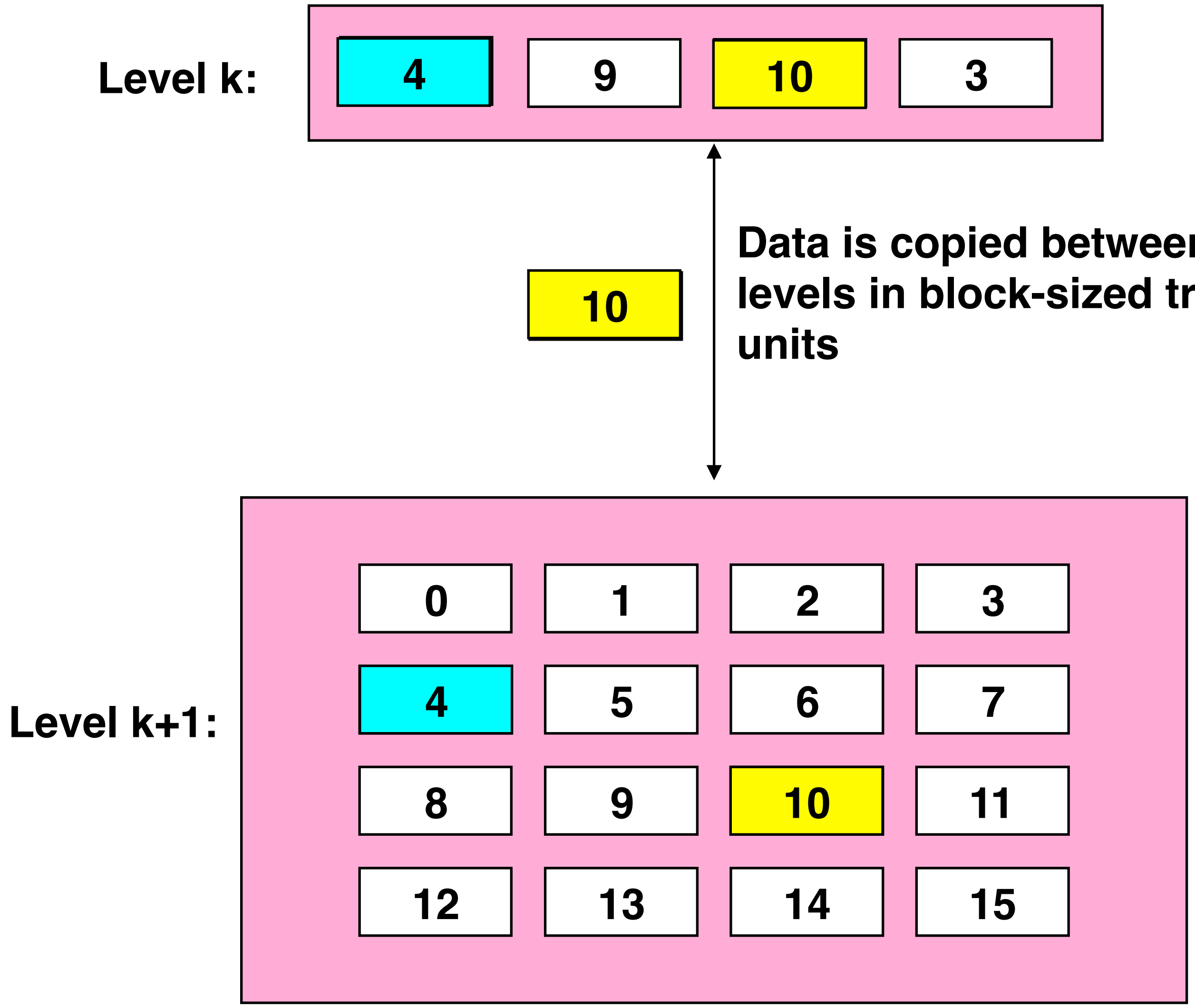
Cache and Memory Hierarchy

- **Cache: A smaller, faster storage device that acts as a staging area for a subset of the data in a larger, slower device.**

- **Fundamental idea of a memory hierarchy:**
  - For each k, the faster, smaller device at level k serves as a cache for the larger, slower device at level k+1.

- **Why do memory hierarchies work?**
  - Programs tend to access the data at level k more often than they access the data at level k+1.
  - Thus, the storage at level k+1 can be slower, and thus larger and cheaper per bit.
  - Net effect: A large pool of memory that costs as much as the cheap storage near the bottom, but that serves data to programs at the rate of the fast storage near the top.

Caching in a Memory Hierarchy

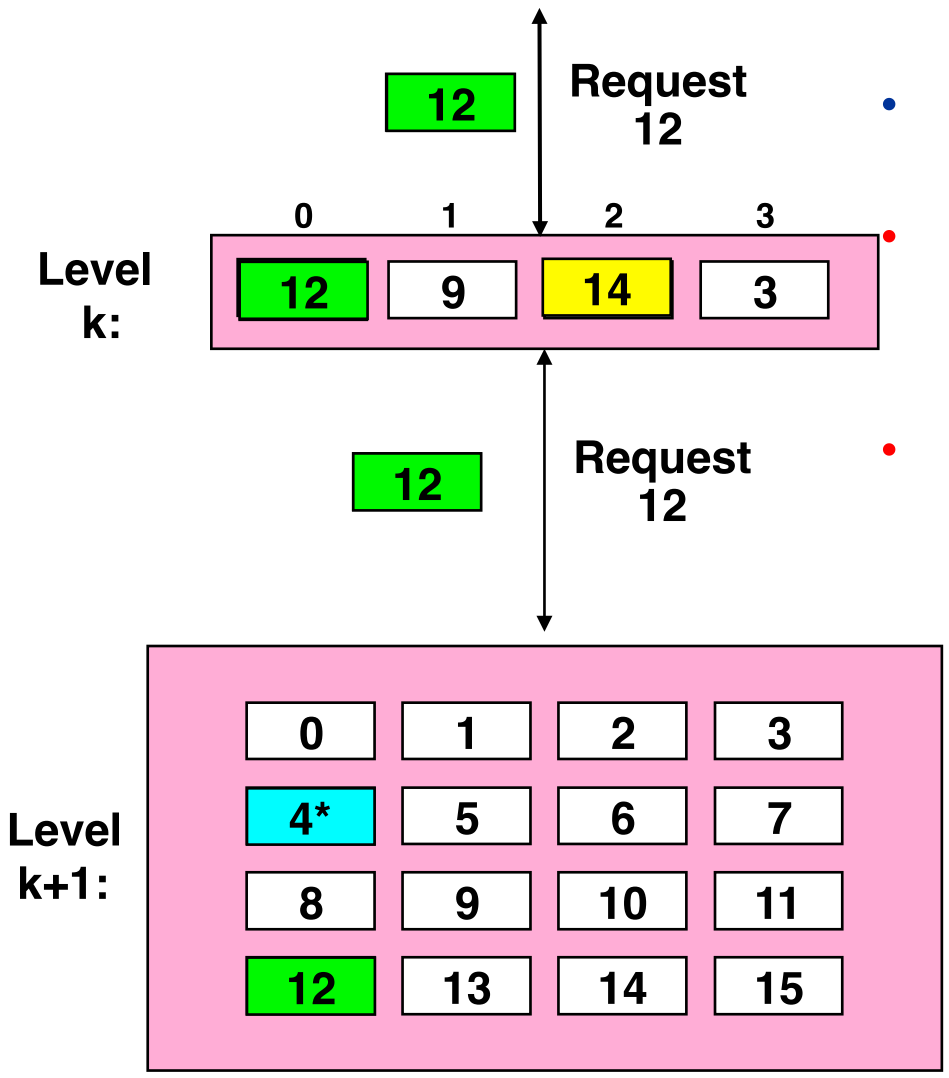


Smaller, faster, more expensive device at level k caches a subset of the blocks from level k+1

Data is copied between levels in block-sized transfer units

Larger, slower, cheaper storage device at level k+1 is partitioned into blocks.

General Caching Concepts



- **Program needs object d, which is stored in some block b.**
- **Cache hit**
  - Program finds b in the cache at level k. E.g., block 14.
- **Cache miss**
  - b is not at level k, so level k cache must fetch it from level k+1. E.g., block 12.
  - If level k cache is full, then some current block must be replaced. Which one is the "victim"?

    Placement policy: where can the new block go? E.g., b mod 4

    Replacement policy: which block should be replaced? E.g., LRU

Caching Miss Types

- **Types of cache misses:**
  - Cold (compulsory) miss

    Cold misses occur because the cache is empty.

  - Conflict miss

    Most caches limit blocks at level k+1 to a small subset (sometimes a singleton) of the block positions at level k.

    E.g. Block i at level k+1 must be placed in block (i mod 4) at level k.

    Conflict misses occur when multiple data objects all map to the same level k block.

    E.g. Referencing blocks 0, 8, 0, 8, 0, 8, ... would miss every time if block 0 and block 8 at level k+1 all map to the same block at level k.

Outline: Pipelining and Memory Hierarchy

– Pipelining Concepts

– Memory Hierarchy Concepts

- **Storage Devices and CPU-Memory Gap**
- **Locality, Cache and Memory Hierarchy**
- **Four Cache and Memory Hierarchy Questions**
- **Practical Memory Hierarchy**

Four Questions for Caches and Memory Hierarchy

- **Q1: Where can a block be placed in the upper level?** ***(Block placement)***
  Required in the exam.

- **Q2: How is a block found if it is in the upper level?** ***(Block identification)***
  Not required in the exam

- **Q3: Which block should be replaced on a miss?** ***(Block replacement)***
  Not required in the exam

- **Q4: What happens on a write?** ***(Write strategy)***
  Not required in the exam

[Patterson]

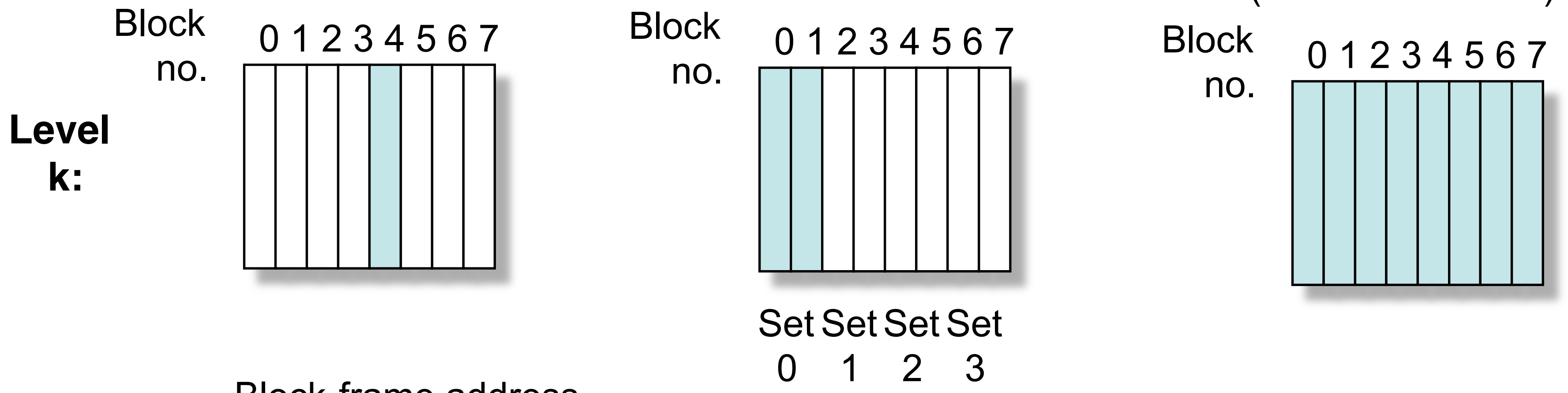
Q1: Where can a block be placed in the upper level?

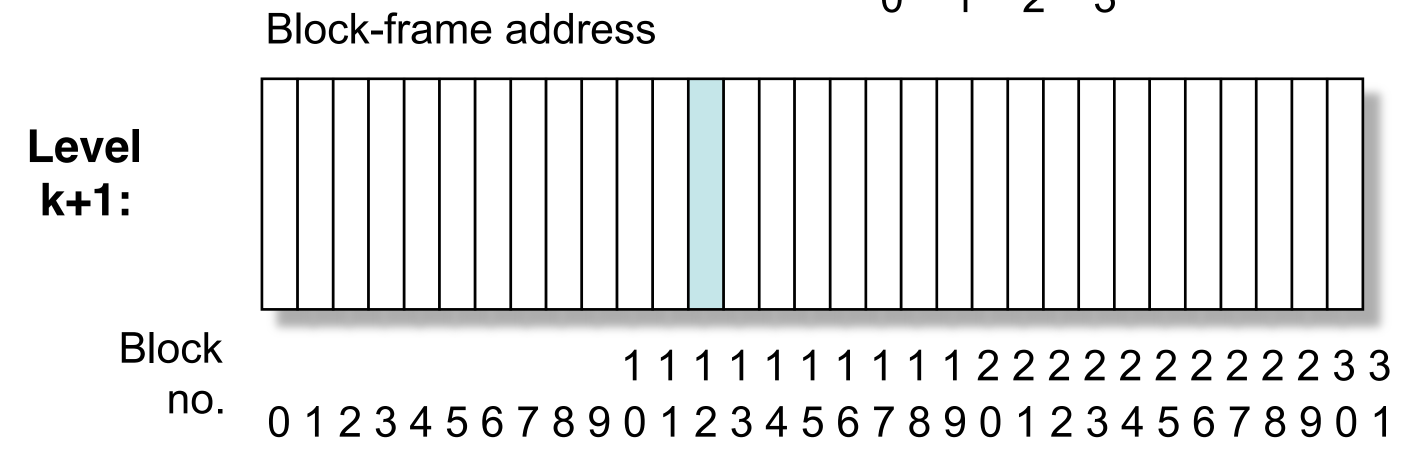
- **Block 12 placed in an 8 block cache:**
  - Fully associative, direct mapped, 2-way set associative
  - Mapped Set Number = Block Number Modulo Number of Sets

Direct mapped:
block 12 can go only into block 4 (12 mod 8 = set 4)

Set associative:
block 12 can go anywhere in set 0 (12 mod 4 = set 0)

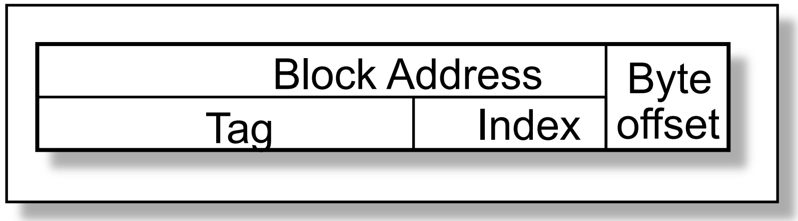
Fully associative:
block 12 can go anywhere in set 0 (12 mod 1 = set 0)

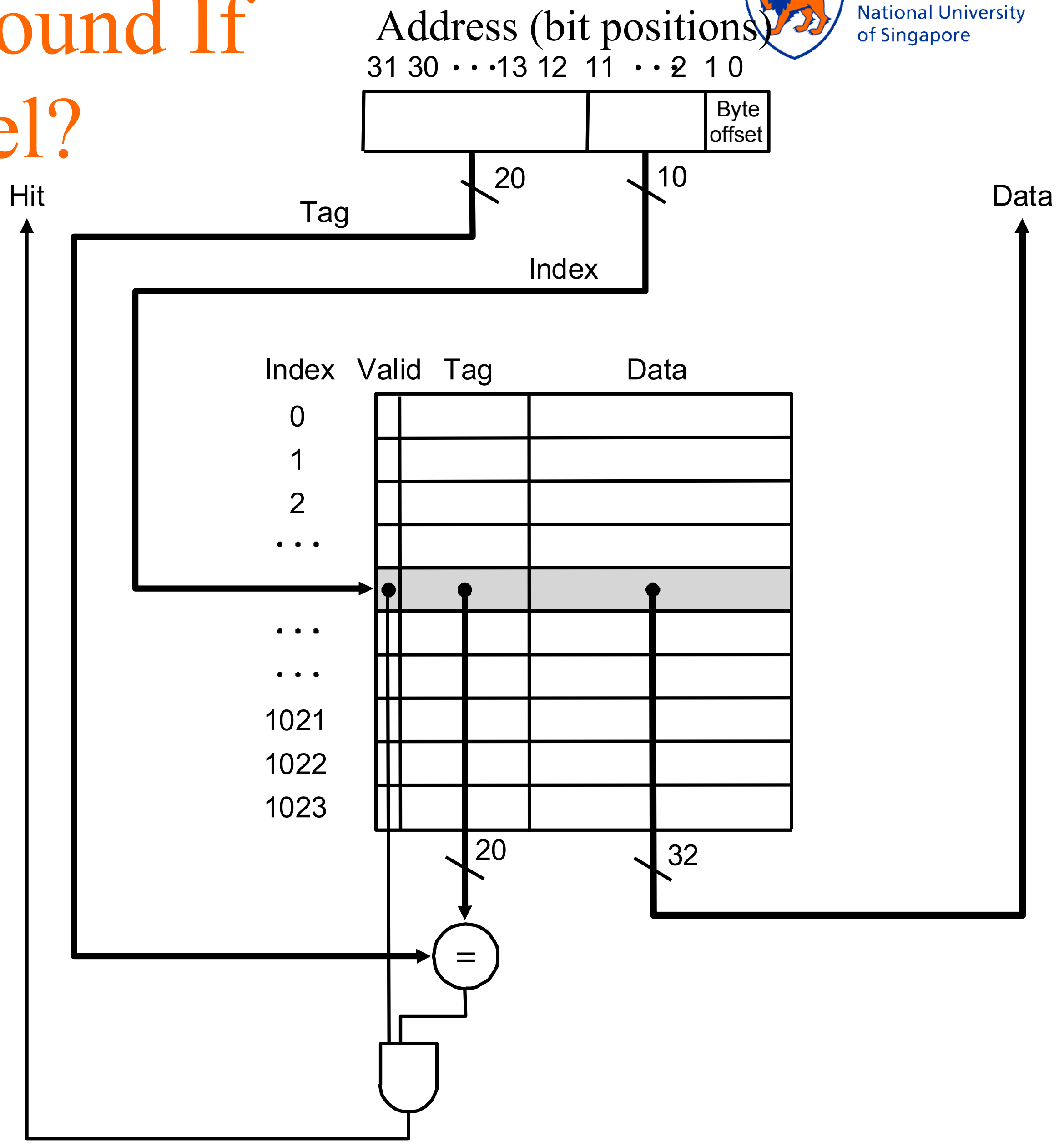




[Patterson]

Q2: How Is a Block Found If It Is In the Upper Level?





- **For each data, check all its possible mapped locations in a cache by comparing its address with the address of the data stored in the locations.**

- **If the comparison is positive, means the wanted data is in the cache. Otherwise, it is a miss.**

Q3: Which Block Should be Replaced on a Miss?

- **Easy for Direct Mapped since there is only one choice**
- **Set Associative or Fully Associative:**
  - Random
  - LRU (Least Recently Used)

Associativity:	**2-way**		**4-way**		**8-way**	
Size	**LRU**	**Random**	**LRU**	**Random**	**LRU**	**Random**
16 KB	**5.2%**	**5.7%**	**4.7%**	**5.3%**	**4.4%**	**5.0%**
64 KB	**1.9%**	**2.0%**	**1.5%**	**1.7%**	**1.4%**	**1.5%**
256 KB	**1.15%**	**1.17%**	**1.13%**	**1.13%**	**1.12%**	**1.12%**

Miss rate comparison when using random or LRU in the replacement algorithm.

Q4: What Happens on a Write?

- ***Write through***—The information is written to both the block in the cache and to the block in the lower-level memory.

- ***Write back***—The information is written only to the block in the cache. The modified cache block is written to main memory only when it is replaced.

- **Pros and Cons of each?**
  - WT: read misses cannot result in writes
  - WB: no writes of repeated writes

- **WT always combined with write buffers so that don't wait for lower level memory**

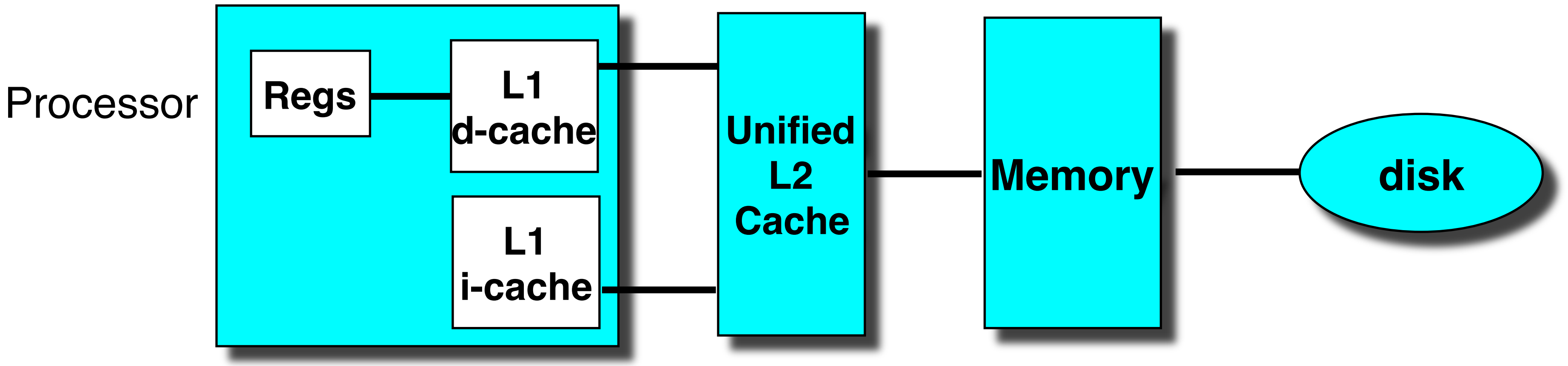
Outline: Pipelining and Memory Hierarchy

– Pipelining Concepts

– Memory Hierarchy Concepts

  - **Storage Devices and CPU-Memory Gap**
  - **Locality, Cache and Memory Hierarchy**
  - **Four Cache and Memory Hierarchy Questions**
  - **Practical Memory Hierarchy**

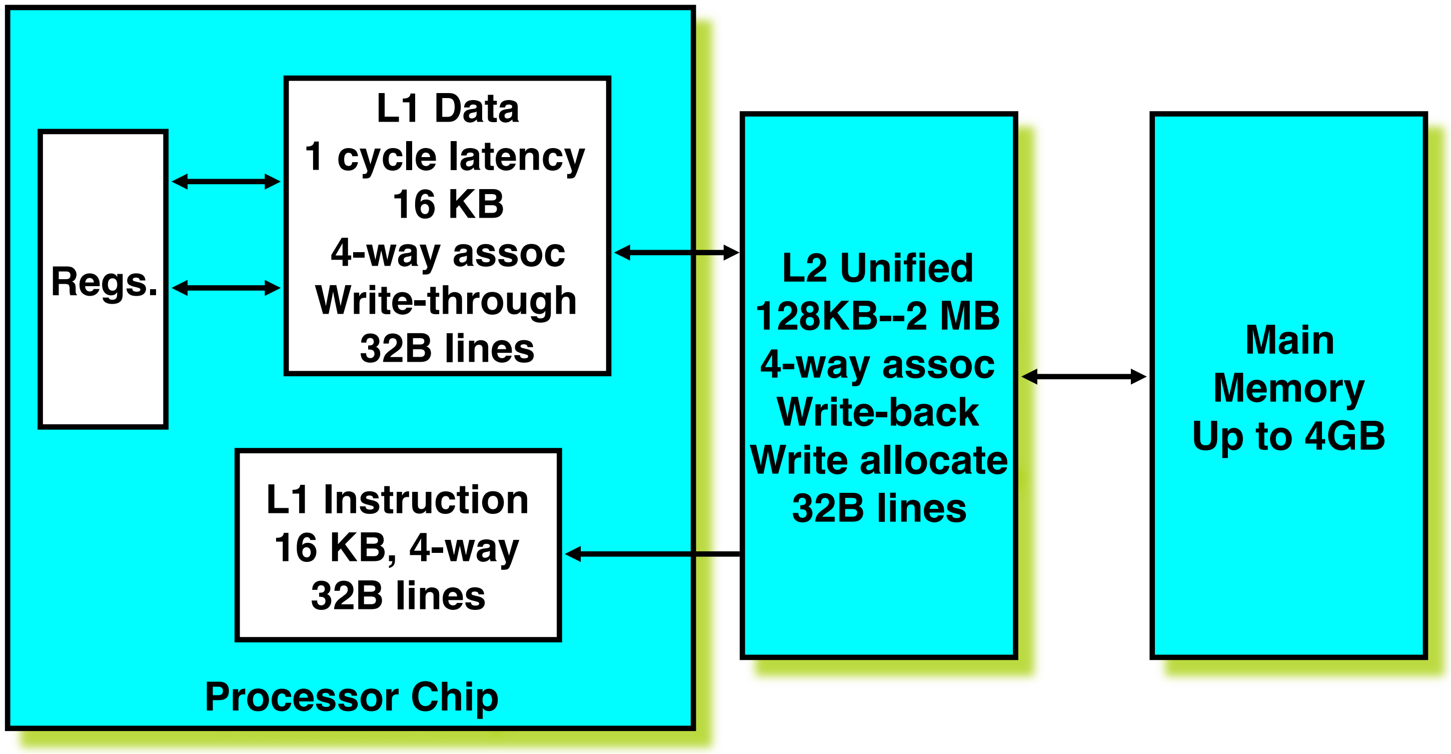
A General Memory Hierarchy



size:	200 B	8-64 KB	1-4MB SRAM	128 MB DRAM	30 GB
speed:	3 ns	3 ns	6 ns	60 ns	8 ms
$/Mbyte:			$100/MB	$1.50/MB	$0.05/MB
line size:	8 B	32 B	32 B	8 KB	

larger, slower, cheaper →

Intel Pentium Memory Hierarchy



Memory Hierarchy Summary

- Different storage devices have different characterizes (such as capacity, access time and cost)
- Locality is a general property of programs
- We want to use memory hierarchy to take advantages of the locality property.
- Store the most frequently used data and instructions in the smallest, fastest but most expensive devices!

Outline: Pipelining and Memory Hierarchy

– Pipelining Concepts

– Memory Hierarch Concepts

– To be covered in the next lecture:

Review for Exam Preparation

Reference book for this lecture notes:

Chapters 6 and 7, *Computer Organization and Design: The Hardware – Software Interface*, Patterson and Hennessy, **3rd edition, ISBN: 1-55860-604-1 , Morgan Kaufmann.**