Faculty Of Computers and Information
Mansoura University

# Hardware of Real-time Systems

Dr. Naira Elazab

Information Technology Dept.,
Faculty Of Computers and Information,
Mansoura University

# Interrupts

- An interrupt is an external hardware signal that initiates an event.
- Interrupts are used to signal that an I/O transfer was completed or needs to be initiated.
- It should be carefully planned which devices or sensors are given a right to interrupt and define its priority.
- The most time-critical events could be recognized with interrupts while others can be recognized with periodic polling (CPU check sensors at a periodic interval).
- Maskable interrupts commonly used for events that occur during regular operating conditions.
- Non-maskable interrupts are reserved for extremely critical events that require immediate action

# Interrupts

- **Masking: preventing the interrupt from disturbing the CPU, the processor will not accept the interrupt signal (interrupts with low priorities).**

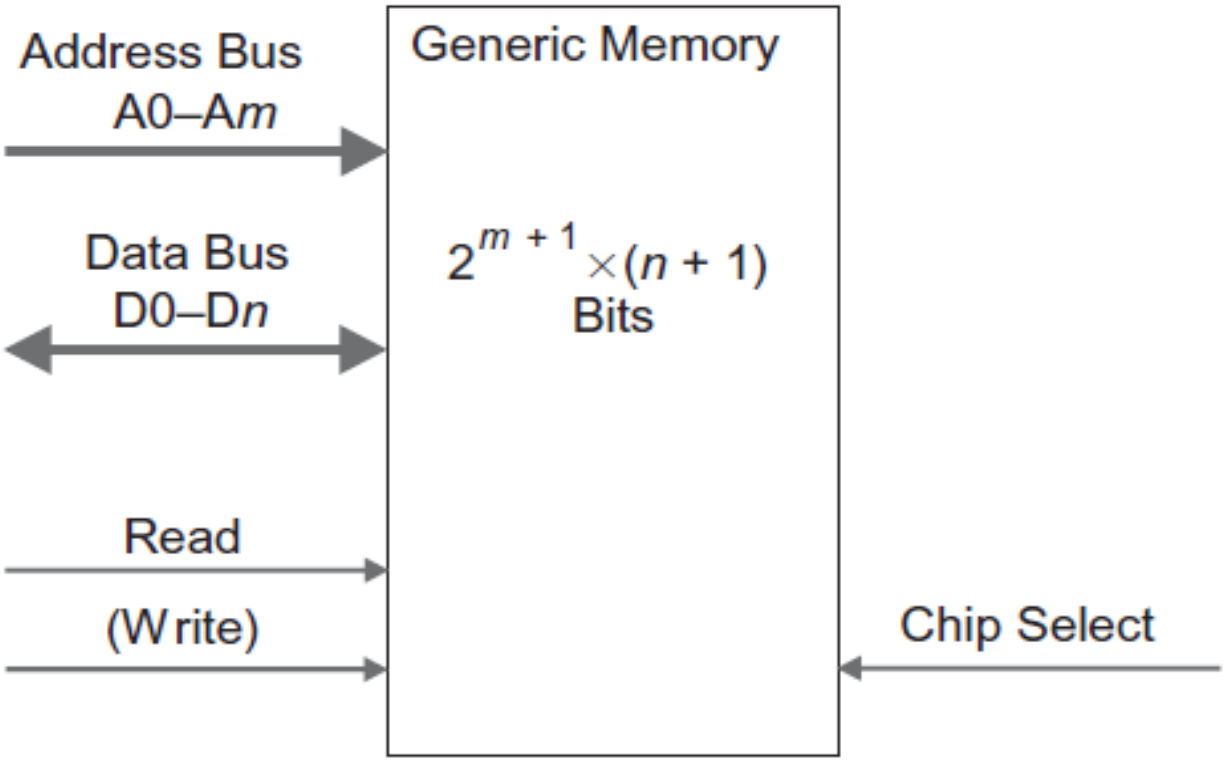
- **Maskable interrupts: a hardware interrupt that may be ignored by changing a bit called mask-bit.**

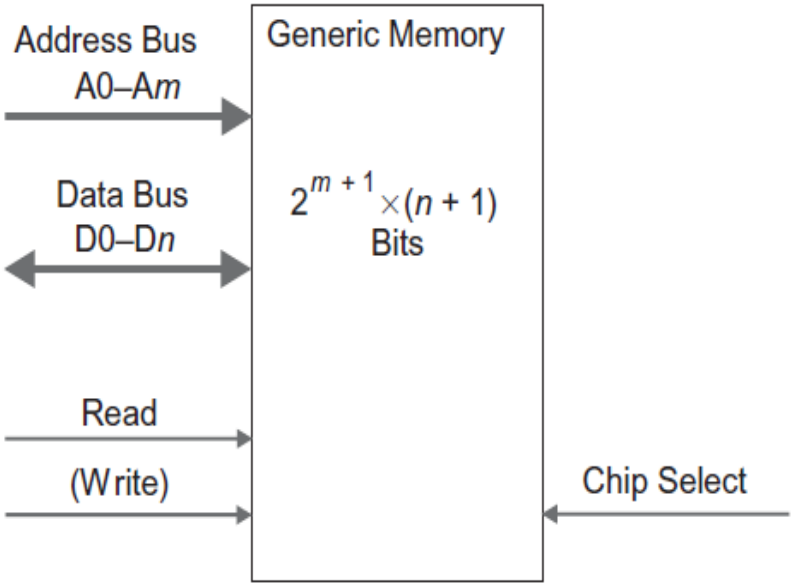
- **Non-maskable interrupts can not be disabled or ignored by CPU.**

# Interrupts

- A typical interrupt service process is as follows:
  - The interrupt - request line is activated.
  - The interrupt request is latched by the CPU hardware
  - The processing of the ongoing instruction is completed
  - The content of program counter register (PCR) is pushed to stack.
  - The content of status register ( SR ) is pushed to stack.
  - The PCR is loaded with the interrupt handler's address.
  - The interrupt handler is executed
  - The original content of SR is popped from stack.
  - The original content of PCR is popped from stack.

# MEMORY TECHNOLOGIES






# MEMORY TECHNOLOGIES

- **Memory Size:** 2 $m$ + 1 × ( $n$ + 1 ) 2 m+1 ×(n+1)
  The memory consists of 2 $m$ + 1 2 m+1 addressable locations, each storing a word of $n$ + 1 n+1 bits.
- **Word Size:** Each word is $n$ + 1 n+1 bits wide, where the bit index $n$ n starts from zero.
- **Number of Words:** The total number of words (memory locations) is 2 $m$ + 1 2 m+1 , where the address index $m$ m starts from zero.
- **Address Bus Size:** The address bus contains $m$ + 1 m+1 lines, allowing access to 2 $m$ + 1 2 m+1 unique memory locations.
- **Data Bus Size:** The data bus width equals the word size, i.e., $n$ + 1 n+1 bits.
- **Control Signals:** The memory unit operates using two primary control signals: Read and Write, which determine the direction of data transfer between the processor and memory.

# DIFFERENT CLASSES OF MEMORY

<table>
<tr><th>Memory Type</th><th>Category</th><th>Erasure</th><th>Write Mechanism</th><th>Volatility</th></tr>
<tr><td>Random-access memory (RAM)</td><td>Read-write memory</td><td>Electrically, byte-level</td><td>Electrically</td><td>Volatile</td></tr>
<tr><td>Read-only memory (ROM)</td><td rowspan="2">Read-only memory</td><td rowspan="2">Not possible</td><td>Masks</td><td rowspan="5">Nonvolatile</td></tr>
<tr><td>Programmable ROM (PROM)</td><td rowspan="4">Electrically</td></tr>
<tr><td>Erasable PROM (EPROM)</td><td rowspan="3">Read-mostly memory</td><td>UV light, chip-level</td></tr>
<tr><td>Electrically Erasable PROM (EEPROM)</td><td>Electrically, byte-level</td></tr>
<tr><td>Flash memory</td><td>Electrically, block-level</td></tr>
</table>

# DIFFERENT CLASSES OF MEMORY

| SRAM | DRAM |
| --- | --- |
| SRAM memory cell is a digital device that uses the same logic elements (flip – flop) used in the processor. | DRAM memory cell can be implemented with a single transistor and capacitor. |
| More expensive | Less expensive |
| Fast | Slow |
| SRAM memory cell is more space intensive | DRAM is more dense (smaller cells more cells per unit area) |
| Does not need a refreshment circuit. | DRAM must be refreshed regularly to avoid any loss of data |
| Cache Memory | Main Memory |

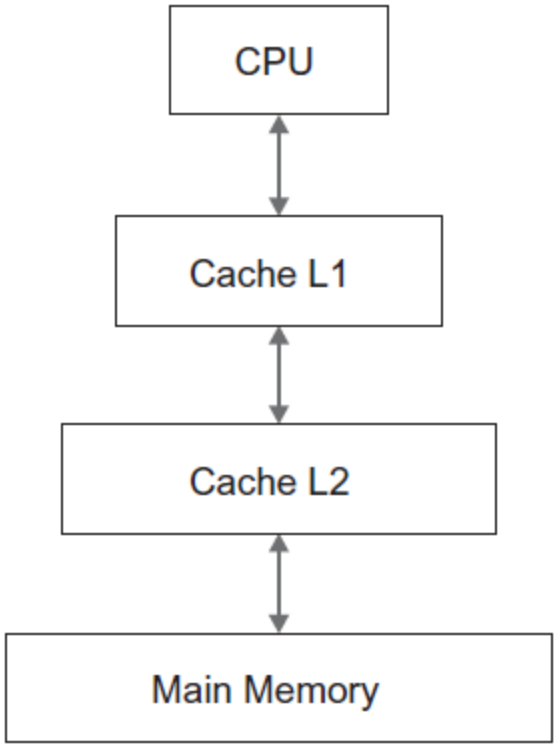
# Different Classes of Memory

- Fast page mode ( FPM ) DRAM
- Extended data output ( EDO ) DRAM
- Synchronous DRAM ( SDRAM )
- Direct Rambus DRAM ( DRDRAM )
- Double data rate 3 synchronous DRAM ( DDR3 SDRAM )
  - DDR4, DDR5, LPDDR5X

# Different Classes of Memory

**Fast Page Mode (FPM) DRAM:**
An early DRAM type that speeds up access within the same memory row by keeping the row open, reducing access time for sequential data.
**Extended Data Output (EDO) DRAM:**
Improved over FPM by allowing the next memory access to start while the previous data is still being output, increasing overall memory throughput.
**Synchronous DRAM (SDRAM):**
Operates in synchronization with the system clock, enabling faster and more predictable data access compared to asynchronous DRAM.
**Direct Rambus DRAM (DRDRAM):**
Uses a high-speed channel architecture to transfer data at very high rates, offering higher bandwidth but at higher cost and complexity.
**Double Data Rate 3 Synchronous DRAM (DDR3 SDRAM):**
Transfers data on both rising and falling edges of the clock signal, offering higher speed and lower power consumption than DDR2.
**DDR4, DDR5, LPDDR5X:**
Successive generations of DDR memory—**DDR4** improves speed and efficiency, **DDR5** doubles bandwidth and enhances reliability, while **LPDDR5X** is a low-power version optimized for mobile and embedded

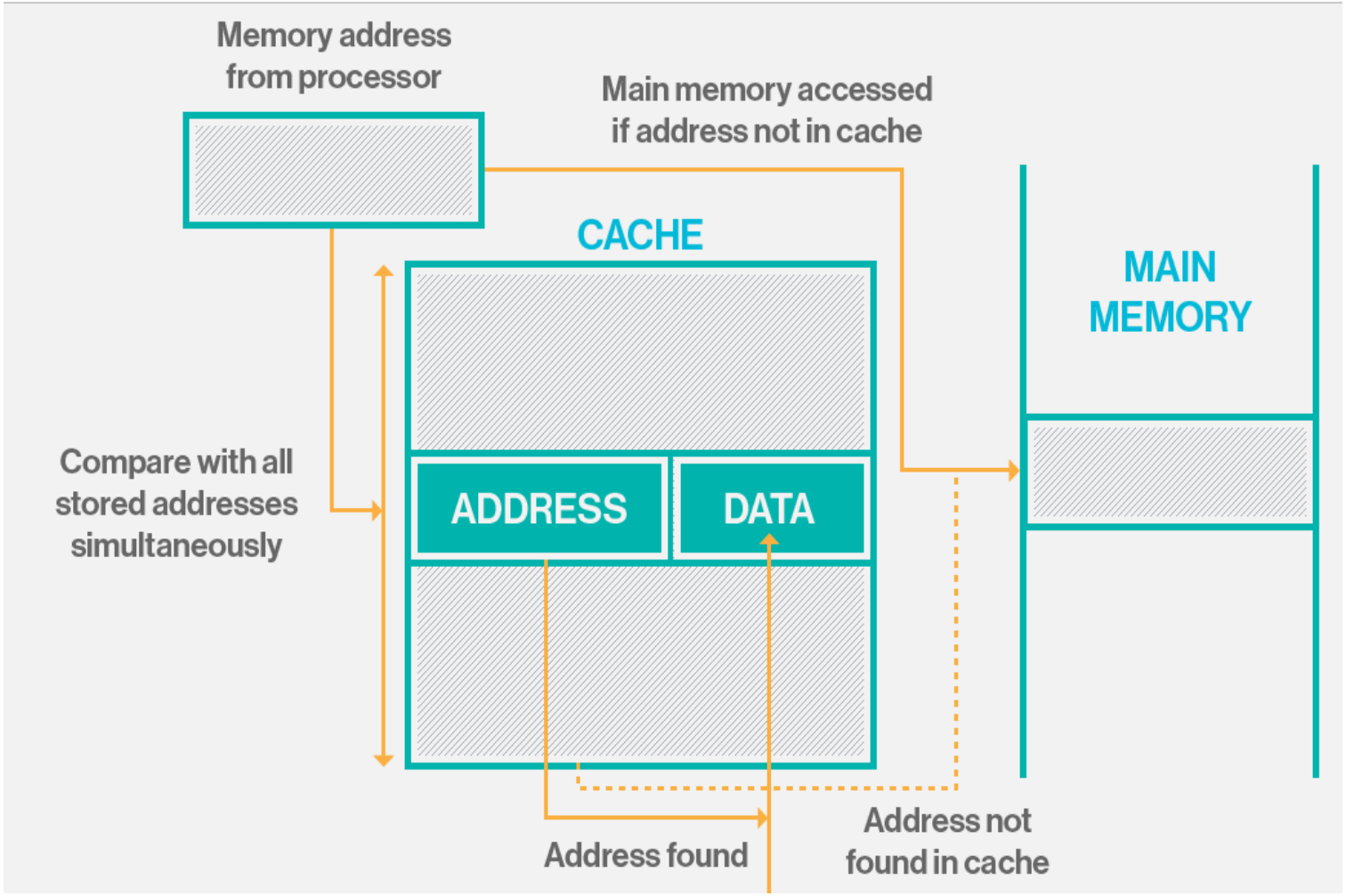
# Hierarchical Memory Organization



# Hierarchical Memory Organization

- A cache is a relatively small storage of fast memory where frequently used instructions and data are kept.
- The cache also contains a list of memory blocks (address tags) that are currently in the cache.
- Performance measure is a function of the cache hit ratio.

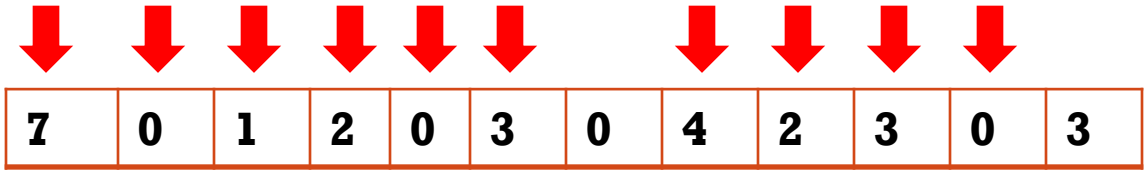
# Cache Memory



# Hierarchical Memory Organization

- **Cache design considerations include:**
  - ✓ **access time:** The time required to retrieve data from the cache, affecting overall system performance and speed.
  - ✓ **cache size:** The total storage capacity of the cache, determining how much data or instructions can be stored for quick access.
  - ✓ **block size:** The amount of data transferred between main memory and cache in one operation, influencing hit rate and transfer efficiency.
  - ✓ **mapping function:** The method used to determine where each memory block is placed in the cache, such as direct, associative, or set-associative mapping.
  - ✓ **Block replacement algorithm:** Determines which cache block should be replaced when new data is loaded, using strategies like **FIFO**, **LRU (Least Recently Used)**, or **Random replacement** to maintain efficiency.
  - ✓ **Write policy:**Defines how data modifications in the cache are handled, typically using **Write-Through** (update both cache and memory immediately) or **Write-Back** (update memory only when the cache block is replaced).
  - ✓ **number of caches (e.g., there can be separate data and instruction caches, or an instruction cache only), and number of cache levels (typically 1 – 3).**

# Cache Memory




| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 7 | 7 | 7 | 2 | 2 | 2 | 4 | 4 | 4 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 |
| | | 1 | 1 | 1 | 3 | 3 | 2 | 2 | 2 |

# Q

- **Consider a two -level memory hierarchy with a single 8 K cache built inside the CPU. Assume a non-cached memory reference costs 100 ns, whereas an access from the cache takes only 20 ns. Now assume that the cache hit ratio is 73%. What would be the average access time for this hierarchy of a system memory.**

# A.

- **Each access is either a hit or a miss, so average memory access time (AMAT) is:**
  **AMAT = time spent in hits + time spent in misses**

  **= hit rate * hit time + miss rate * miss time**

# Q

- **Consider a three - level memory hierarchy with an 8 K upper level cache built inside the CPU and an external 128 K lower level cache. Assume the access times of 20 and 60 ns, respectively, and the cost of non-cached memory reference is 100 ns. The upper level hit rate is again 73%, and the lower level hit rate is 89%. What would be the average access time for this hierarchy of a system memory.**

# A.

- **Each access is either a hit or a miss, so average memory access time (AMAT) is:**

AMAT =

rel hit rate L1 * hit time L1 + rel miss rate L1 * (rel hit rate L2 * hit time L2 + rel miss rate L2 * (hit time RAM))