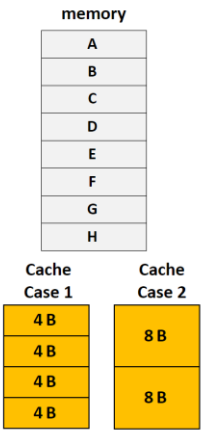
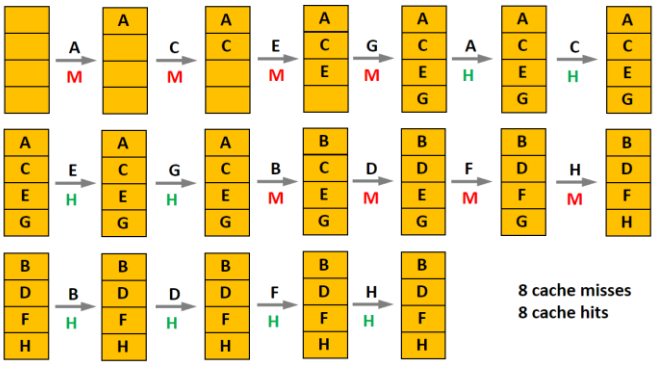
Problem 3

- Assume the following configuration for a processor with a cache:
  - Cache size = 16B
  - Cache block size = 4B or 8B
  - Data variable size = 4B
  - Allocation of data variables in the main memory =>
  - *Evicting the earliest installed item* when cache is full
- The following access sequence for data variables is generated by a program
  - **A C E G A C E G B D F H B D F H**
- Please
  - Compute the number of cache hits and misses for the two different cache block sizes (4B or 8B)
  - Explain the results to evaluate **cache block size design**



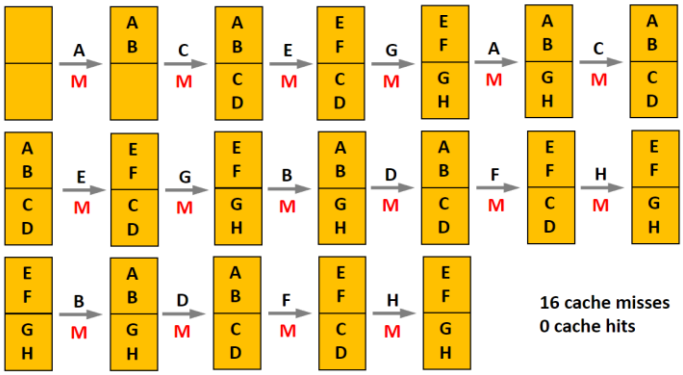
Solution to Problem 3

- **A C E G A C E G B D F H B D F H**
- Case 1: block size = 4B

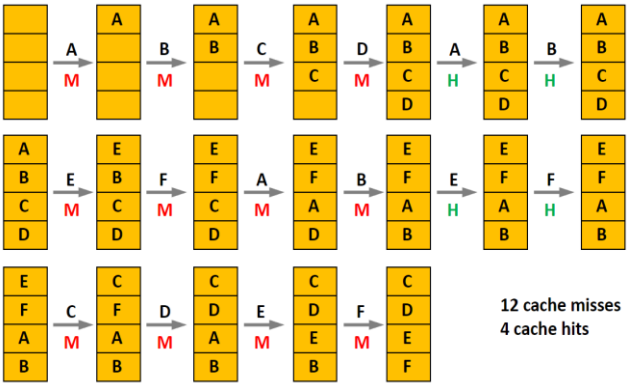


Solution to Problem 3

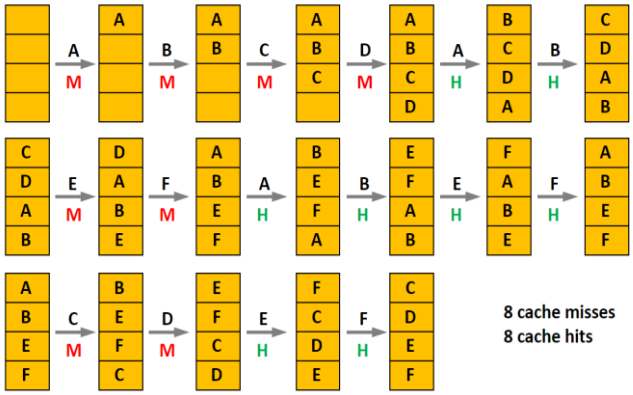
- **A C E G A C E G B D F H B D F H**
- Case 2: block size = 8B



- **A B C D A B E F A B E F C D E F**
- FIFO replacement: evict the earliest installed



- **A B C D A B E F A B E F C D E F**
- LRU replacement



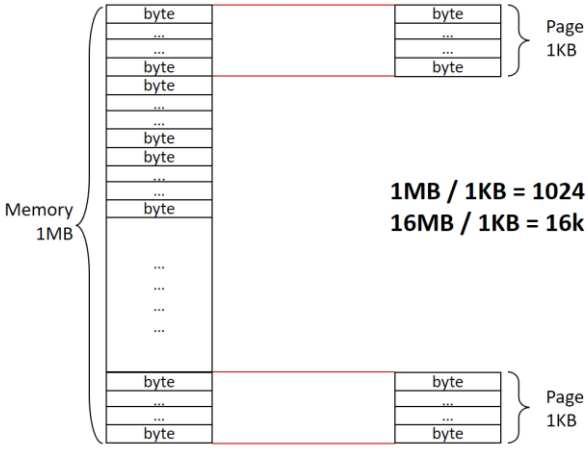
Problem 2: address translation

- A main memory (hardware) has size 1MB
- A program (virtual memory) has size 16MB
- In a paging system, the size of one page is 1KB

- Questions
  - How many frames does the main memory contain?
  - How many virtual pages does the program contain?

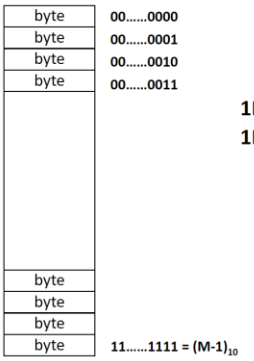
  - How many bits are needed to address a byte in the main memory?
  - How many bits are needed to address a byte in the virtual memory?
  - How many bits are needed to address a byte in a page?

Problem 2: address translation



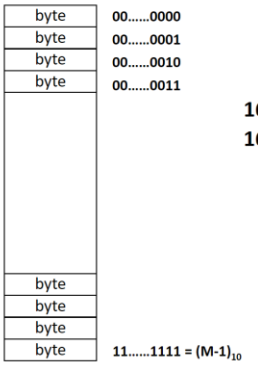
1MB / 1KB = 1024
16MB / 1KB = 16k

- **How many bits are needed to address a byte in the *main memory*?**



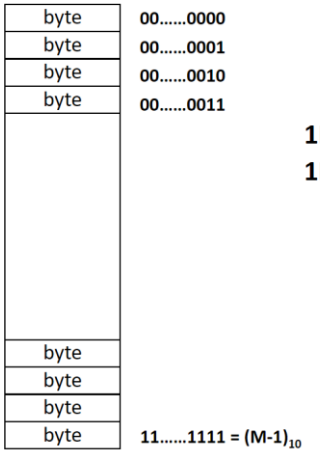
$1M = 2^{20}$
1M → 20 bits

- **How many bits are needed to address a byte in the *virtual memory*?**



$16M = 2^{24}$
16M → 24 bits

- **How many bits are needed to address a byte in a *page*?**



$1K = 2^{10}$
1K → 10 bits