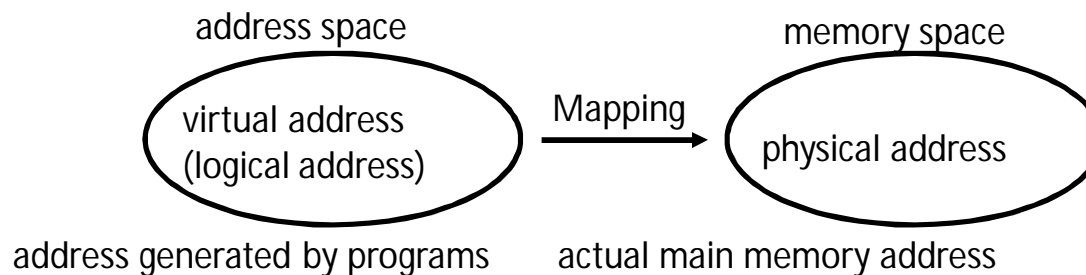# Virtual memory

- Large memory space were available equal to the totality of auxiliary memory.
- VM system provides mechanism for translating program-generated address into correct main memory locations . This is done dynamically while pgm are being executed in CPU.
- **ADDRESS SPACE AND MEMORY SPACE**

   An address used by programmer is called virtual address and the set of such address is called address space.
- **An** address in main memory is called a location or physical address. Set of such locations is called memory space.
- Address space is the set of address generated by programs as they reference inst and data.
- Memory space consist of actual main memory locations directly addressable for processing .
- Address space is allowed to be larger than memory space in computer with virtual memory
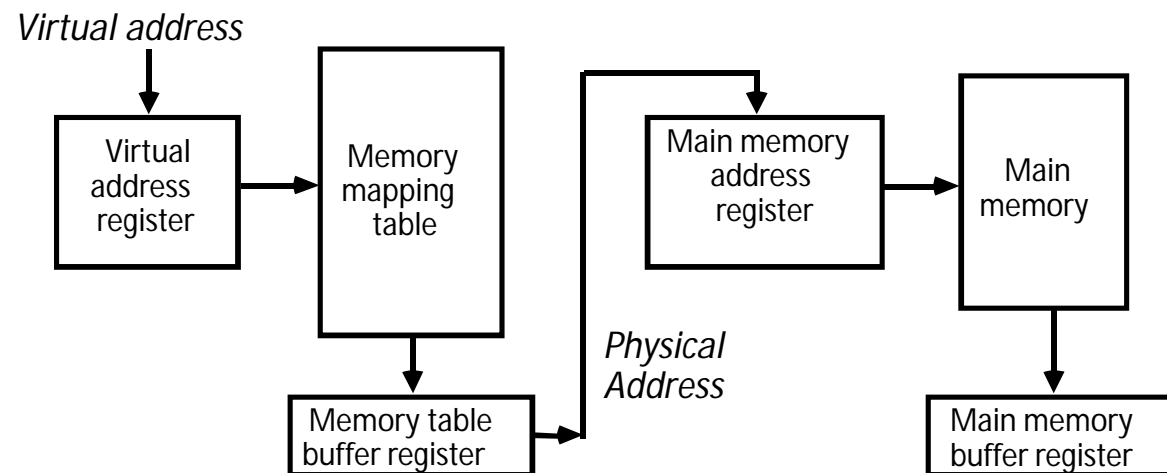
VIRTUAL MEMORY

> Give the programmer the illusion that the system has a very large memory,
> even though the computer actually has a relatively small main memory

Address Space(Logical) and Memory Space(Physical)

address space   memory space

(virtual address (logical address)) → Mapping → (physical address)

address generated by programs   actual main memory address

Address Mapping

Memory *Mapping Table* for *Virtual Address -> Physical Address*

Virtual address

| Virtual address register | → | Memory mapping table | | Main memory address register | → | Main memory |

Physical Address

| Memory table buffer register | | Main memory buffer register |

ADDRESS MAPPING

Address Space and Memory Space are each divided
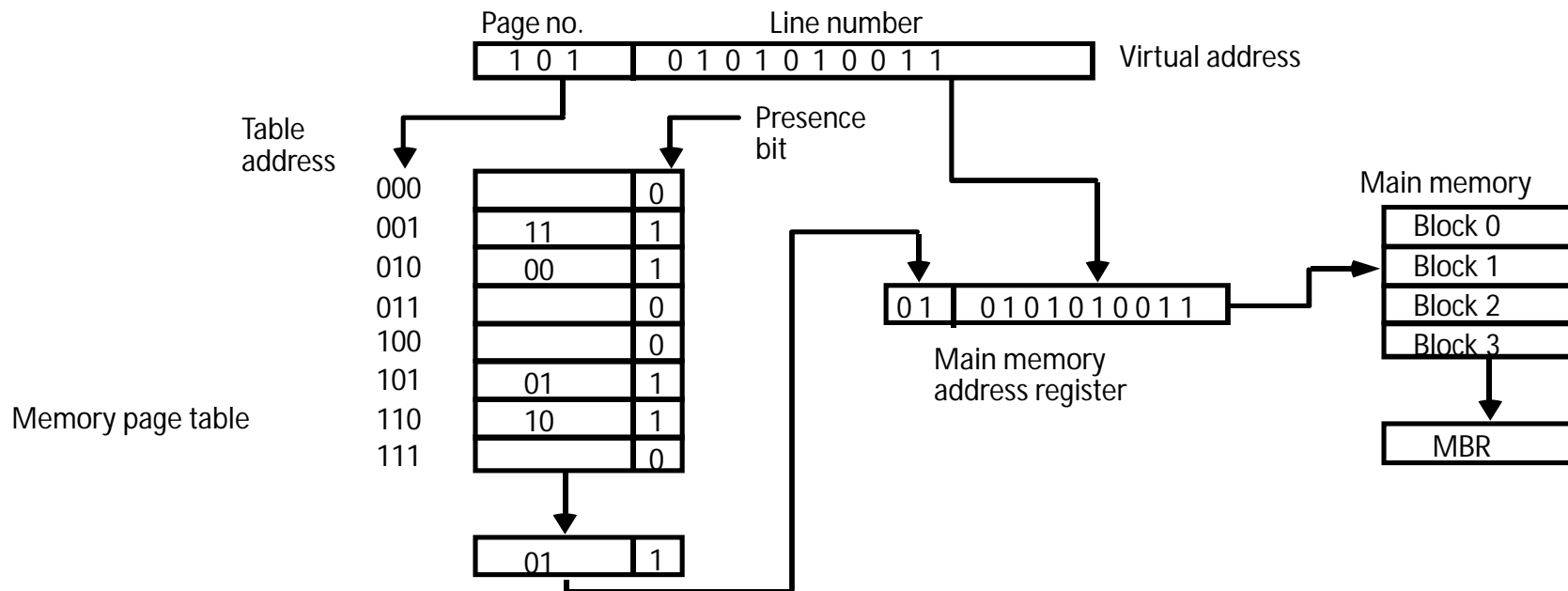into fixed size group of words called *blocks* or *pages*

1K words group

Page 0
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7

Address space
N = 8K = 2^{13}

Memory space
M = 4K = 2^{12}

Block 0
Block 1
Block 2
Block 3

Organization of memory Mapping Table in a paged system

Page no.   Line number

| 1 0 1 | 0 1 0 1 0 1 0 0 1 1 | Virtual address

Table
address

Presence
bit

000		0
001	11	1
010	00	1
011		0
100		0
101	01	1
110	10	1
111		0

Memory page table

| 0 1 | 0 1 0 1 0 1 0 0 1 1 |

Main memory
address register

| 01 | 1 |

Main memory

Block 0
Block 1
Block 2
Block 3

| MBR |

ASSOCIATIVE MEMORY PAGE TABLE

Assume that
      Number of Blocks in memory = m
      Number of Pages in Virtual Address Space = n

Page Table
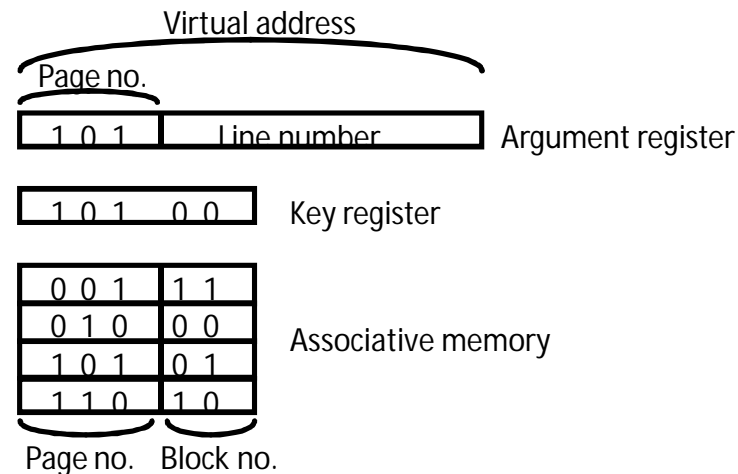   - Straight forward design -> n entry table in memory
      Inefficient storage space utilization
      <- n-m entries of the table is empty

    - More efficient method is m-entry Page Table
        Page Table made of an Associative Memory
        m words; (Page Number:Block Number)

Virtual address

Page no.

| 1 0 1 | Line number | Argument register |

| 1 0 1 | 0 0 | Key register |

0 0 1	1 1
0 1 0	0 0
1 0 1	0 1
1 1 0	1 0

Associative memory

Page no.   Block no.

Page Fault
      Page number cannot be found in the Page Table