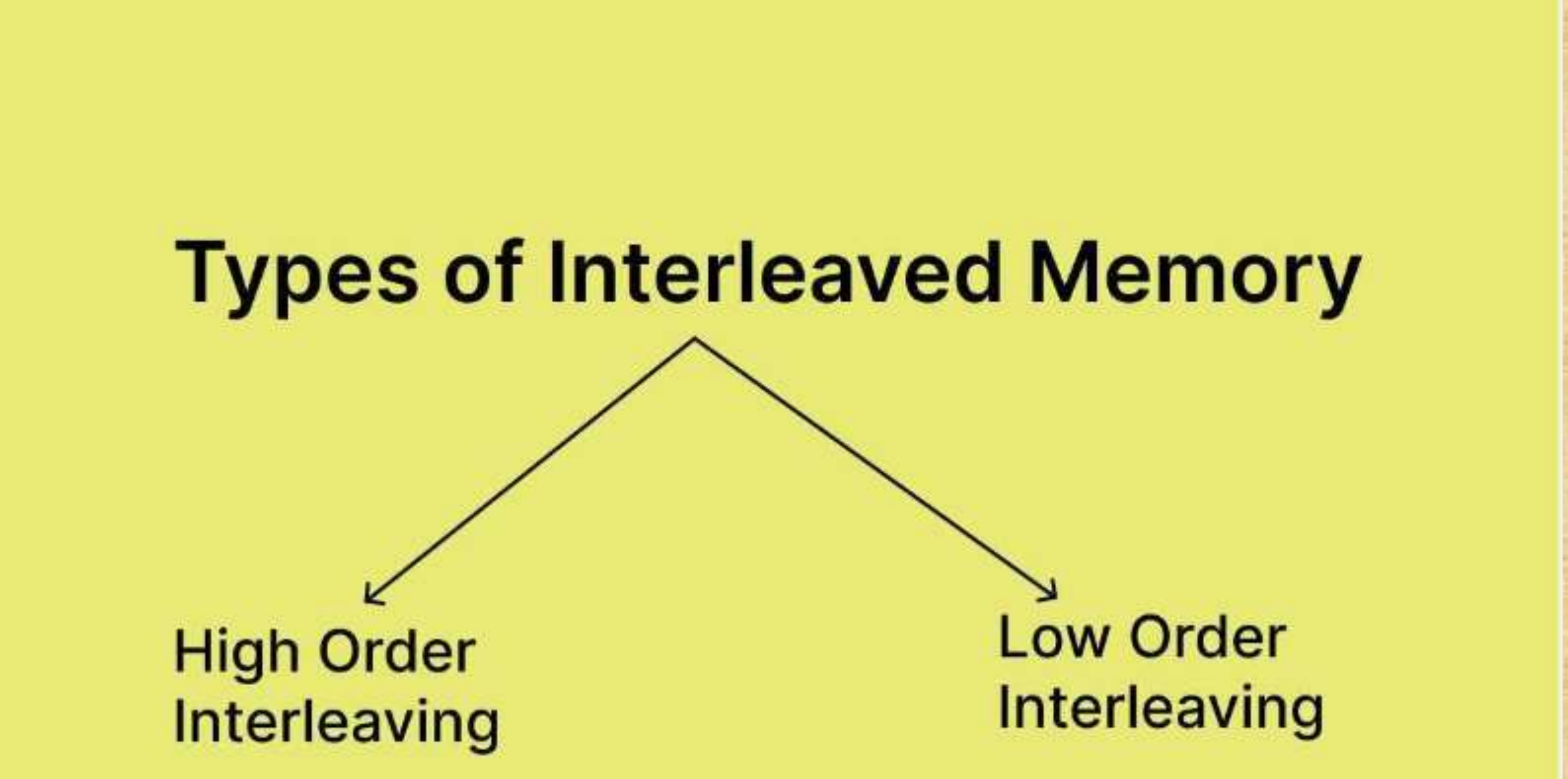
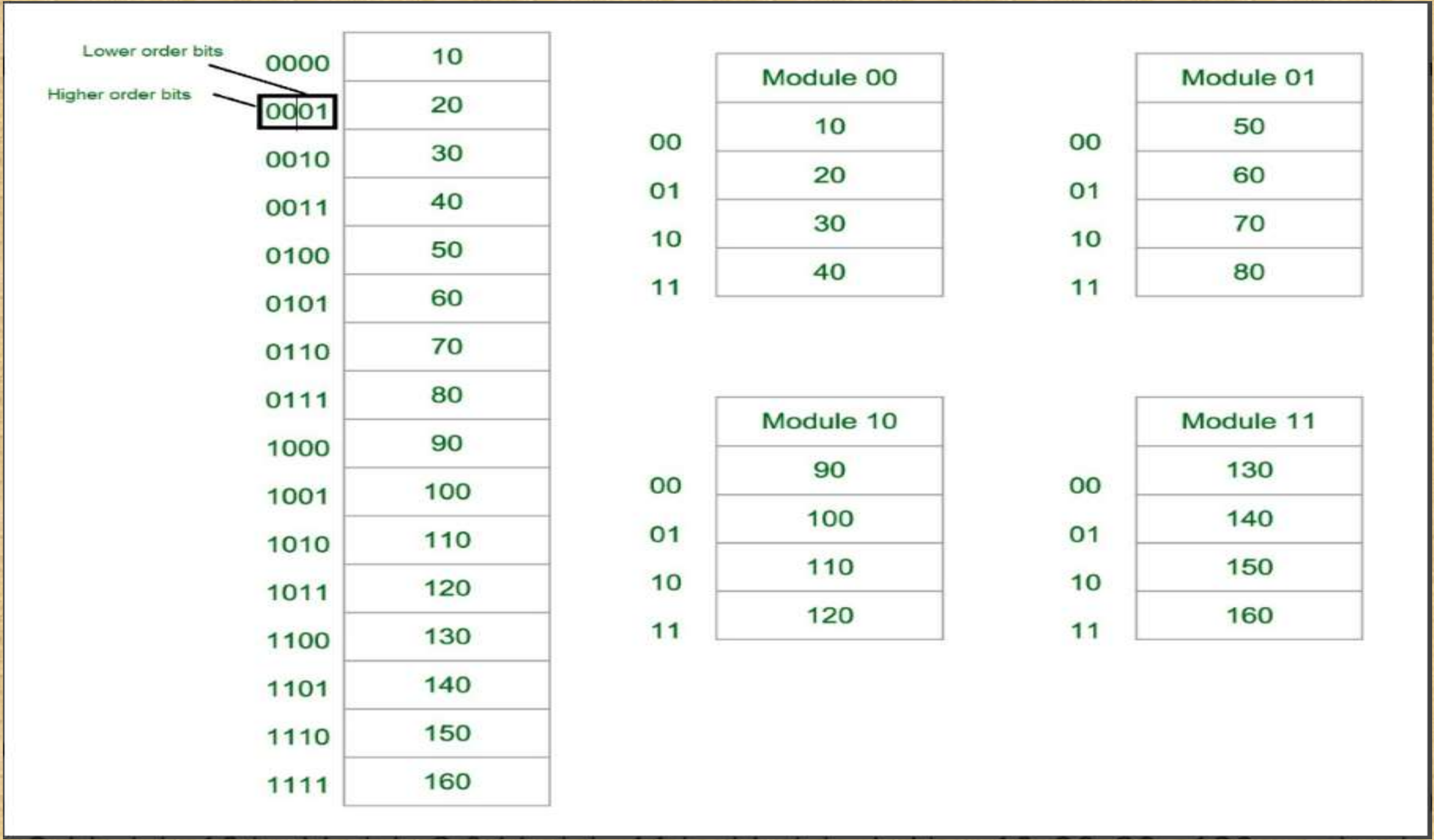
Interleaved Memory Operation

- It is a Technique that divides memory into a number of modules such that Successive words in the address space are placed in the Different modules.



Interleaved Memory Operation



0000	10
0001	20
0010	30
0011	40
0100	50
0101	60
0110	70
0111	80
1000	90
1001	100
1010	110
1011	120
1100	130
1101	140
1110	150
1111	160

	Module 00
00	10
01	20
10	30
11	40

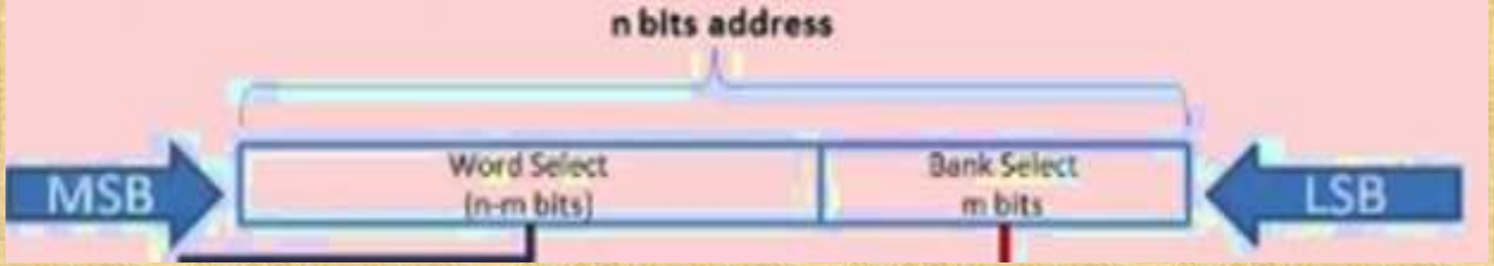
	Module 01
00	50
01	60
10	70
11	80

	Module 10
00	90
01	100
10	110
11	120

	Module 11
00	130
01	140
10	150
11	160

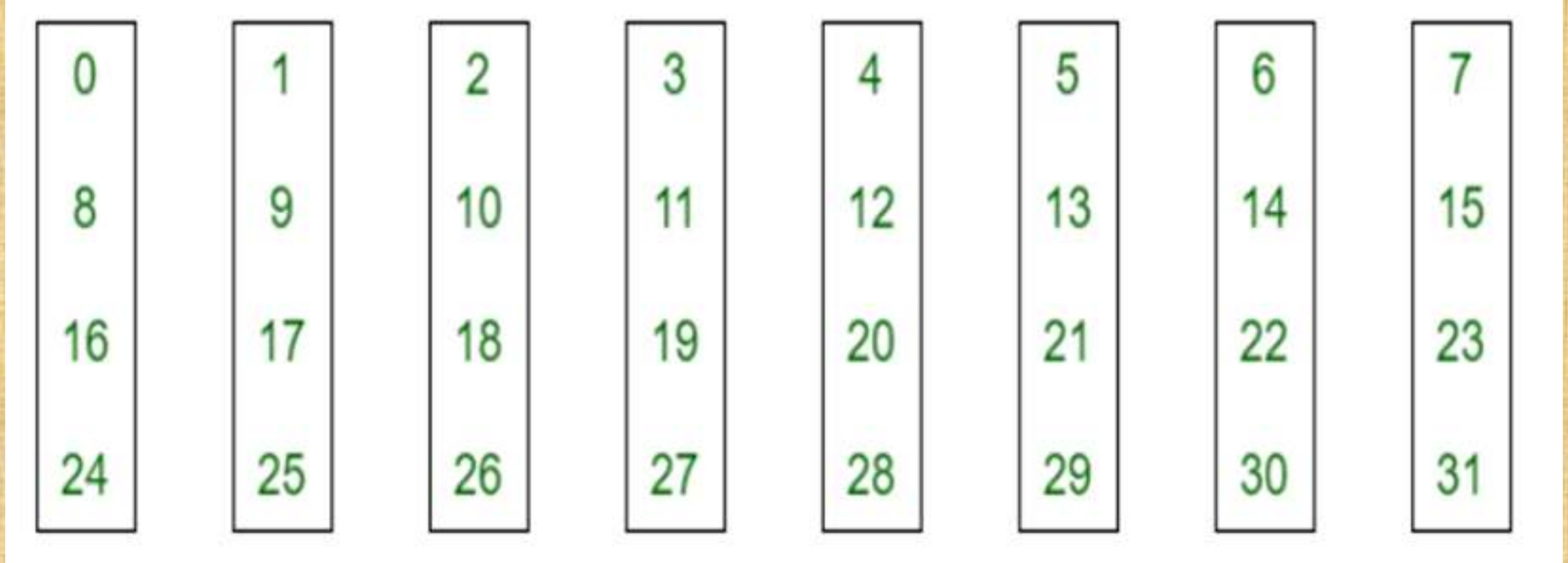
Interleaved Memory Operation

- **Low Order Interleaving**
- It uses the least significant bits of the memory address to determine which memory banks store specific data.
- It divides memory into multiple banks and assigns sequential memory locations to consecutive banks.
- This facilitates parallel access to various data sets, as accessing consecutive addresses results in accessing different memory banks.
- It improves memory access time and bandwidth and enhances overall system performance.



Interleaved Memory Operation

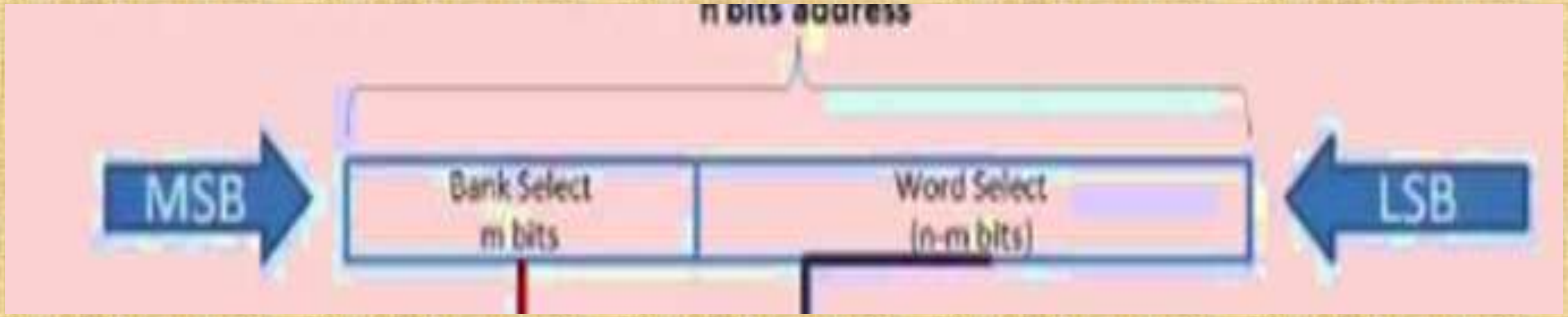
- Low Order Interleaving



0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

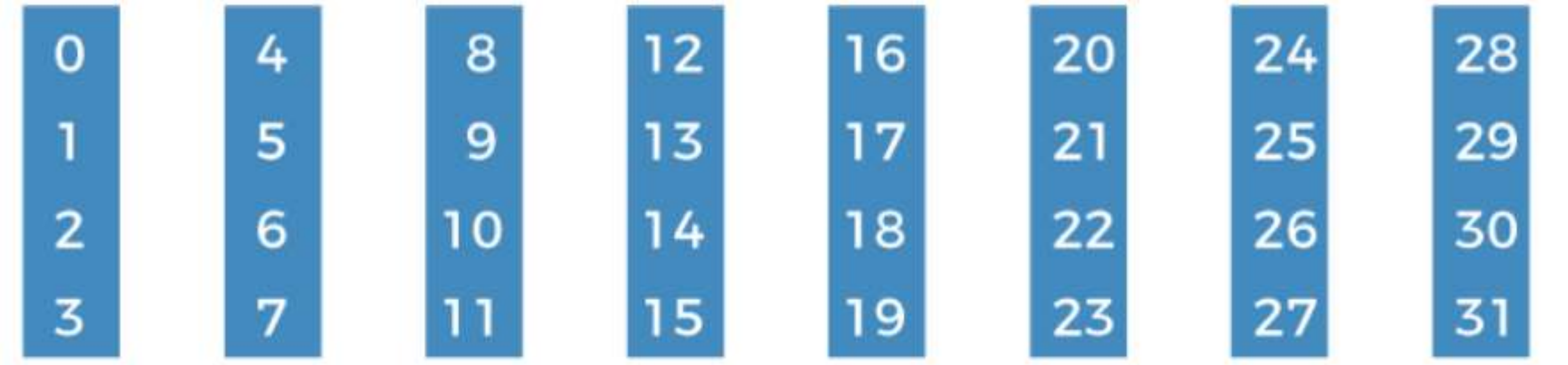
Interleaved Memory Operation

- High Order Interleaving
- It uses the most significant bits of the memory address to determine which memory banks store specific data.
- It divides memory into multiple banks and assigns consecutive memory locations to the same bank.
- This allows for efficient access to related data, as accessing consecutive addresses results in accessing the same memory module.
- It improves memory access time and bandwidth and enhances overall system performance.



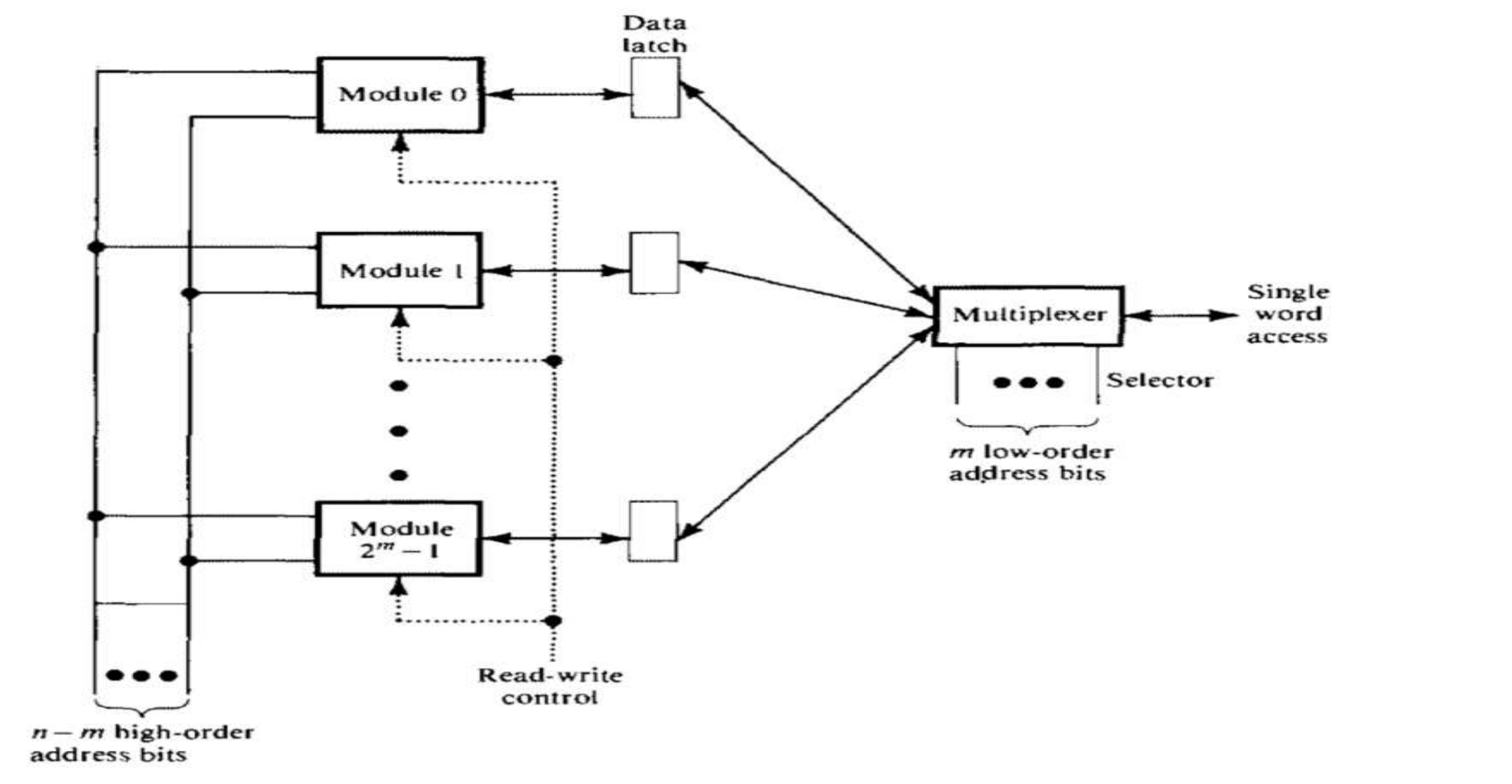
Interleaved Memory Operation

- ❖ High Order Interleaving



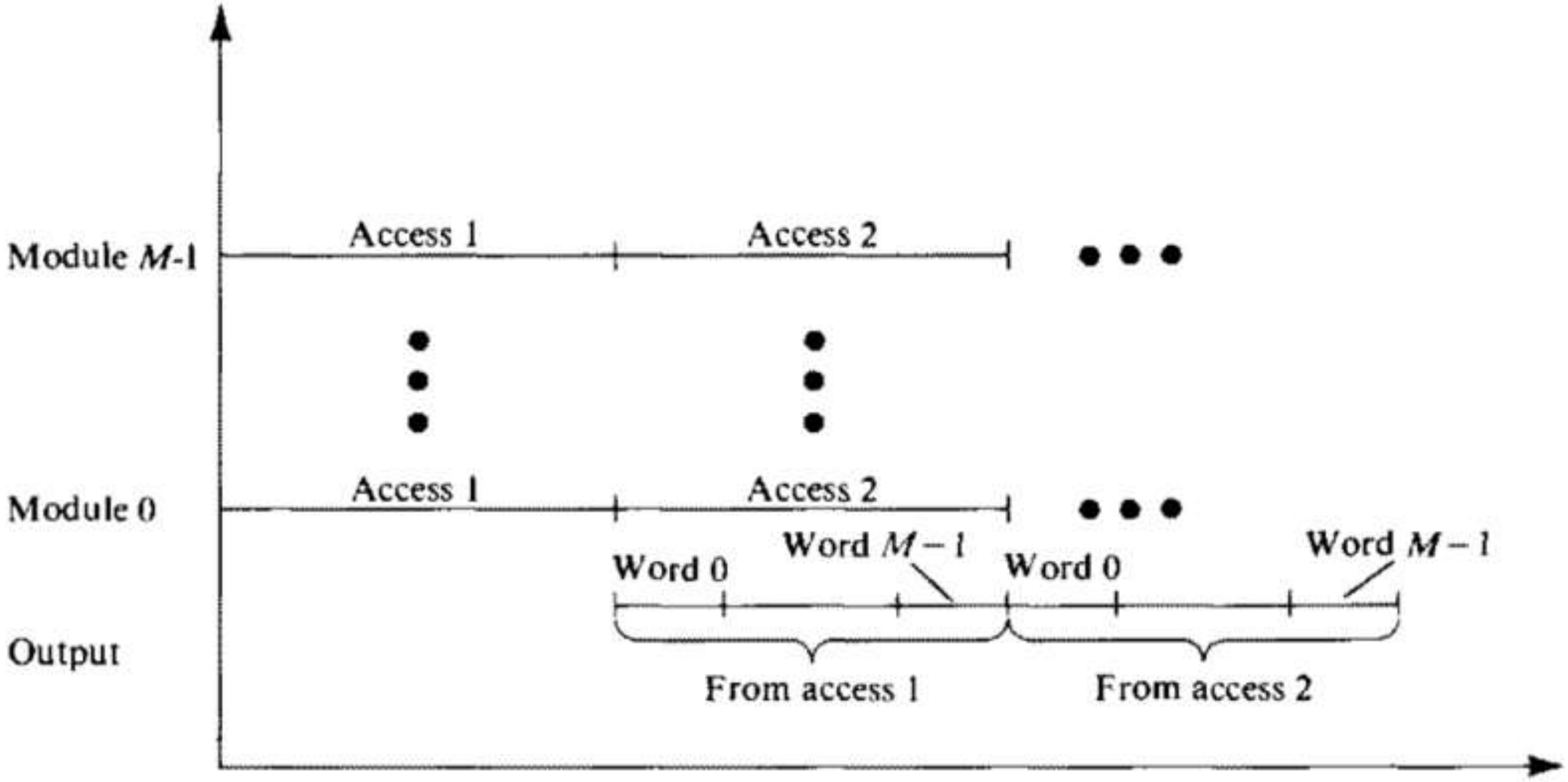
0	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

S-Access Memory Configuration



(*a*) *S*-access memory configuration

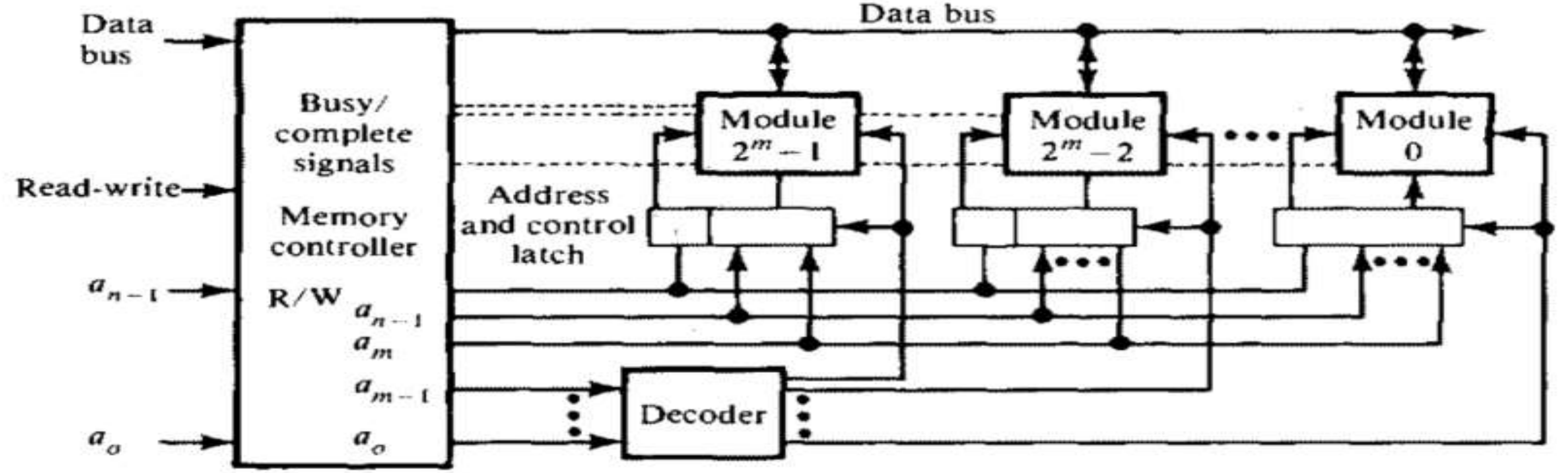
S-Access Memory Configuration



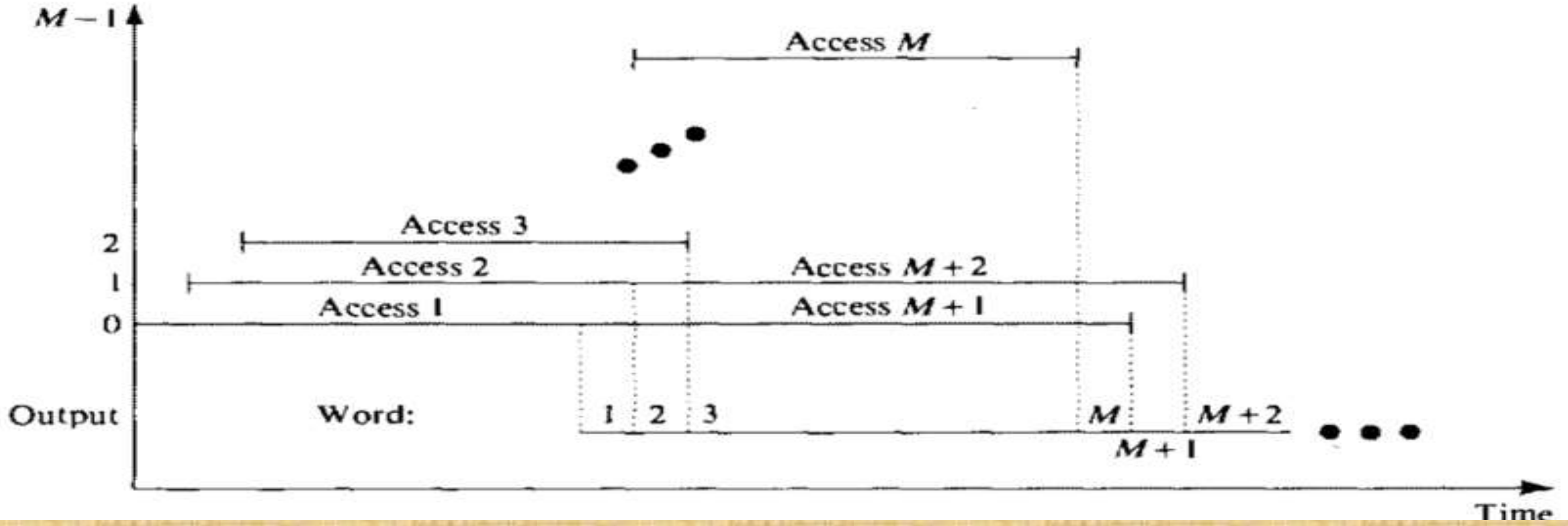
(*b*) Timing diagram for *S*-access configuration

Figure 3.8 The S-access interleaved memory configuration.

C-Access Memory Configuration (Concurrent Access)



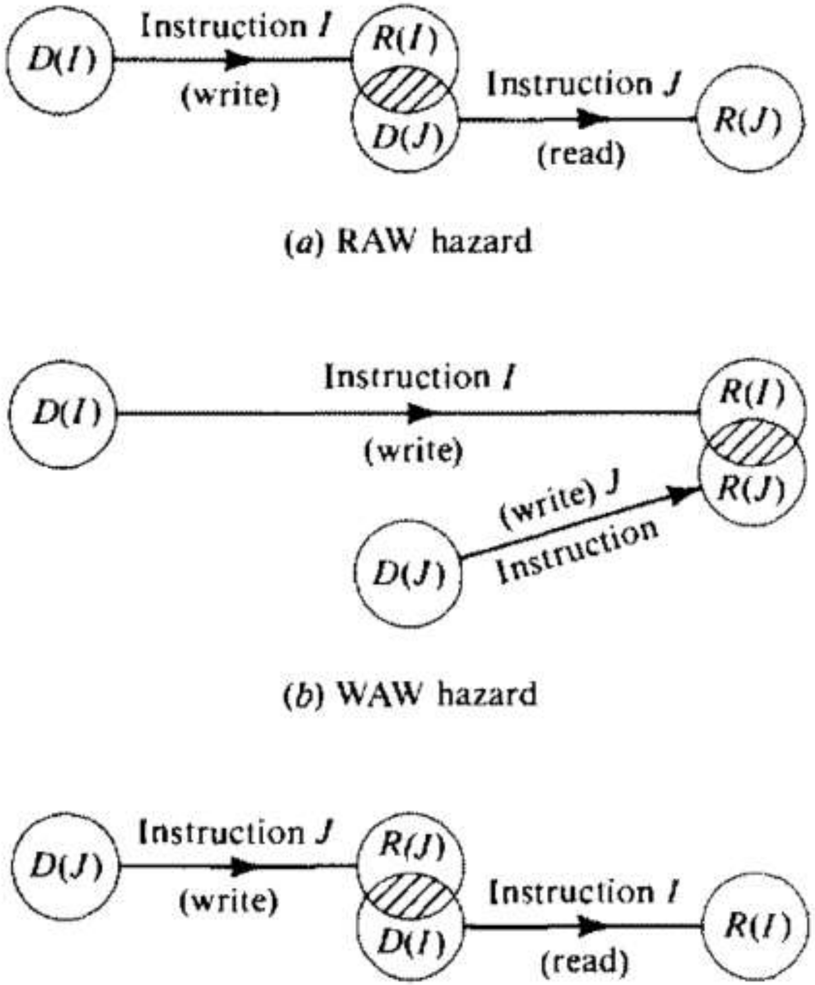
(*a*) *C*-access memory configuration



Hazard Detection and Resolution

- There are four types of data dependencies:
- Read after Write (RAW):j tries to read a source before i writes it, so j incorrectly gets the old value.,
- Write after Read (WAR):j tries to write a destination before it is read by i, so i incorrectly gets the new value.,
- Write after Write (WAW), and
- Read after Read (RAR).

Hazard Detection and Resolution



(*a*) RAW hazard

(*b*) WAW hazard

(*c*) WAR hazard

Figure 3.36 Illustration of RAW, WAW, and WAR hazard conditions.