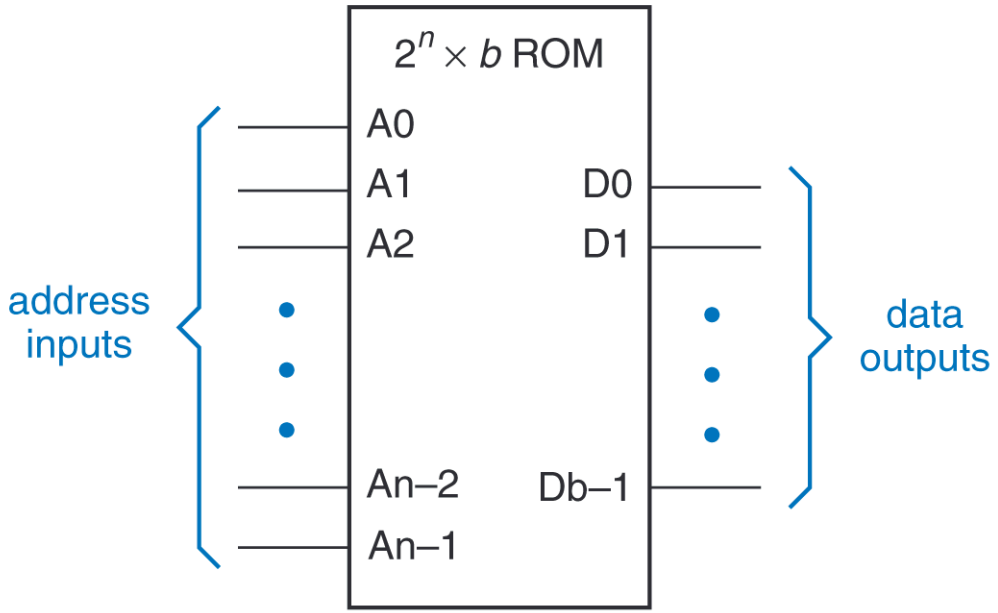


Figure 9-1

Basic structure of a $2^n \times b$ ROM.

Inputs			Outputs			
A2	*A1*	*A0*	*D3*	*D2*	*D1*	*D0*
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Table 9-1

Truth table for a 3-input, 4-output combinational logic function.

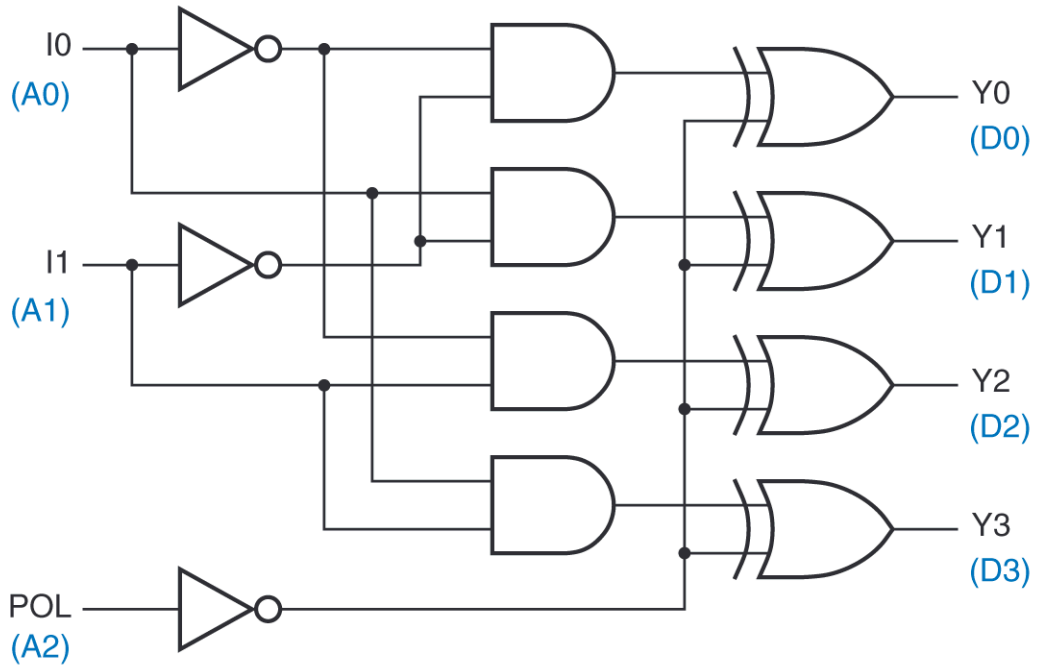


Figure 9-2

A 2-to-4 decoder with output-polarity control.

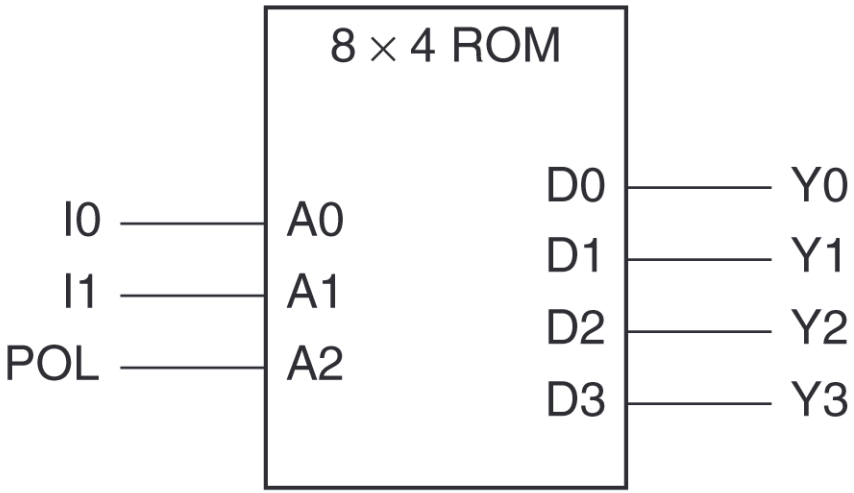


Figure 9-3

Connections to build the 2-to-4 decoder using an 8 × 4 ROM that stores Table 9-1.

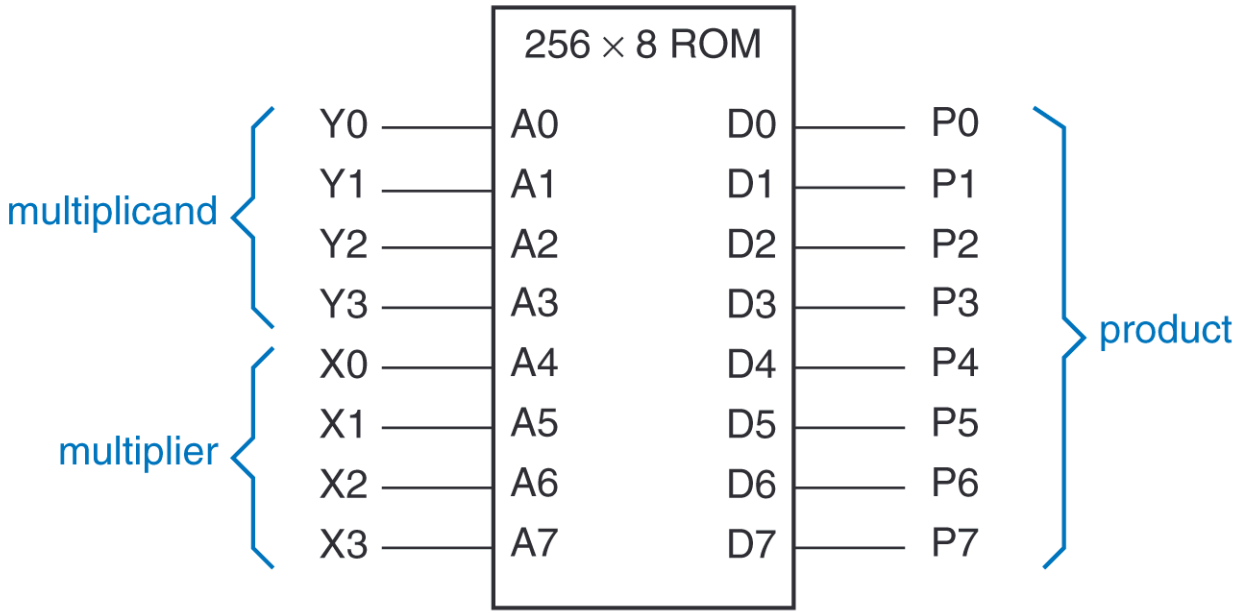


Figure 9-4

Connections to perform a 4×4 unsigned binary multiplication using a 256×8 ROM.

```
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
20: 00 02 04 06 08 0A 0C 0E 10 12 14 16 18 1A 1C 1E
30: 00 03 06 09 0C 0F 12 15 18 1B 1E 21 24 27 2A 2D
40: 00 04 08 0C 10 14 18 1C 20 24 28 2C 30 34 38 3C
50: 00 05 0A 0F 14 19 1E 23 28 2D 32 37 3C 41 46 4B
60: 00 06 0C 12 18 1E 24 2A 30 36 3C 42 48 4E 54 5A
70: 00 07 0E 15 1C 23 2A 31 38 3F 46 4D 54 5B 62 69
80: 00 08 10 18 20 28 30 38 40 48 50 58 60 68 70 78
90: 00 09 12 1B 24 2D 36 3F 48 51 5A 63 6C 75 7E 87
A0: 00 0A 14 1E 28 32 3C 46 50 5A 64 6E 78 82 8C 96
B0: 00 0B 16 21 2C 37 42 4D 58 63 6E 79 84 8F 9A A5
C0: 00 0C 18 24 30 3C 48 54 60 6C 78 84 90 9C A8 B4
D0: 00 0D 1A 27 34 41 4E 5B 68 75 82 8F 9C A9 B6 C3
E0: 00 0E 1C 2A 38 46 54 62 70 7E 8C 9A A8 B6 C4 D2
F0: 00 0F 1E 2D 3C 4B 5A 69 78 87 96 A5 B4 C3 D2 E1
```

Table 9-3

Hexadecimal text file specifying the contents of a 4×4 multiplier ROM.

```
#include <stdio.h>

/* Procedure to print d as a hex digit. */
void PrintHexDigit(int d)
{
    if (d<10) printf("%c", '0'+d);
    else printf("%c", 'A'+d-10);
}

/* Procedure to print i as two hex digits.  */
void PrintHex2(int i)
{
    PrintHexDigit((i / 16) % 16);
    PrintHexDigit(i % 16);
}

void main()
{
  int x, y;

  for (x=0; x<=15; x++) {
    PrintHex2(x*16); printf(":");
    for (y=0; y<=15; y++) {
      printf(" ");
      PrintHex2(x*y);
    }
    printf("\n");
  }
}
```

Table 9-4

Program to generate the text file specifying the contents of a 4×4 multiplier ROM.

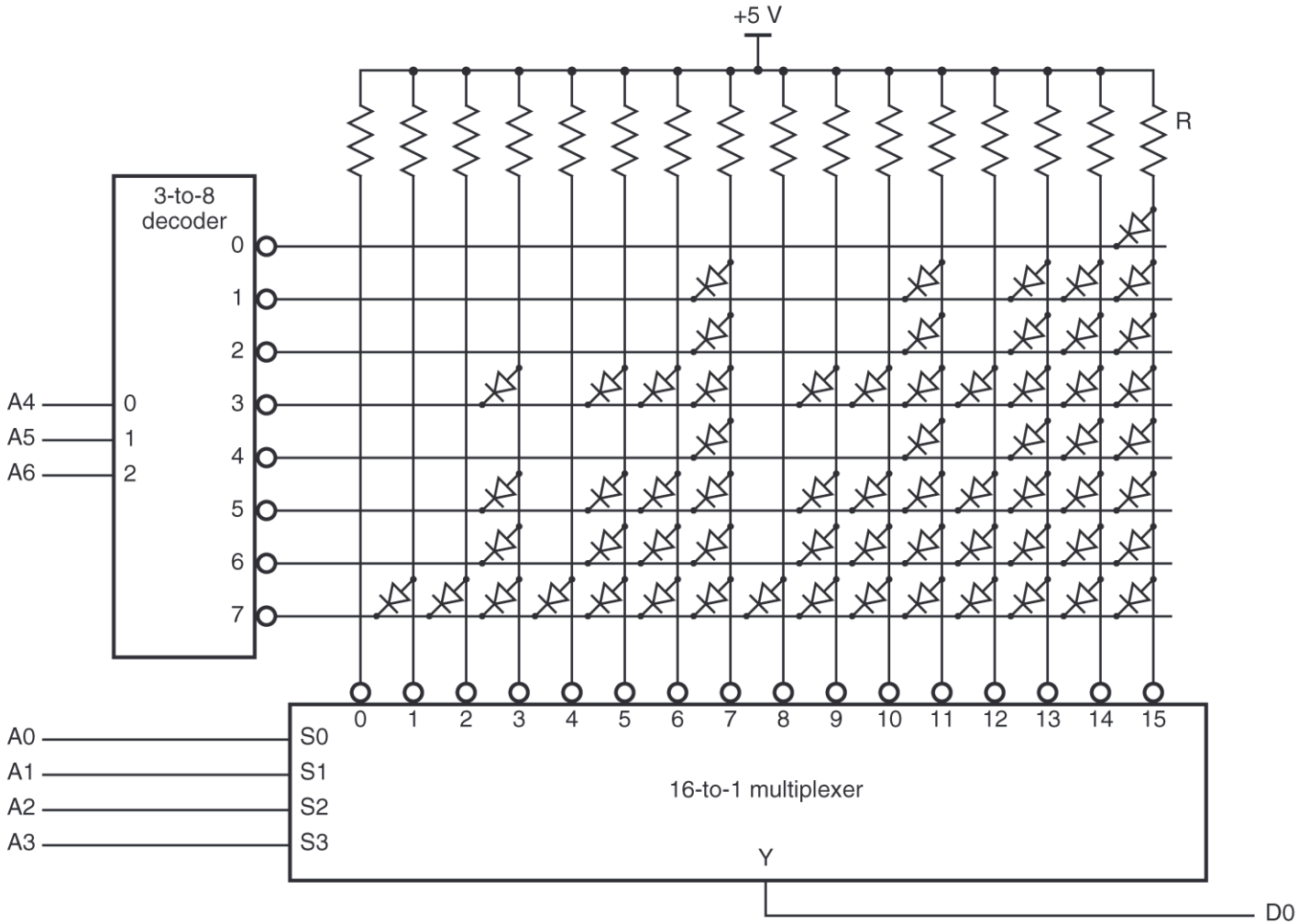


Figure 9-7

Internal structure of a 128 × 1 ROM using two-dimensional decoding.

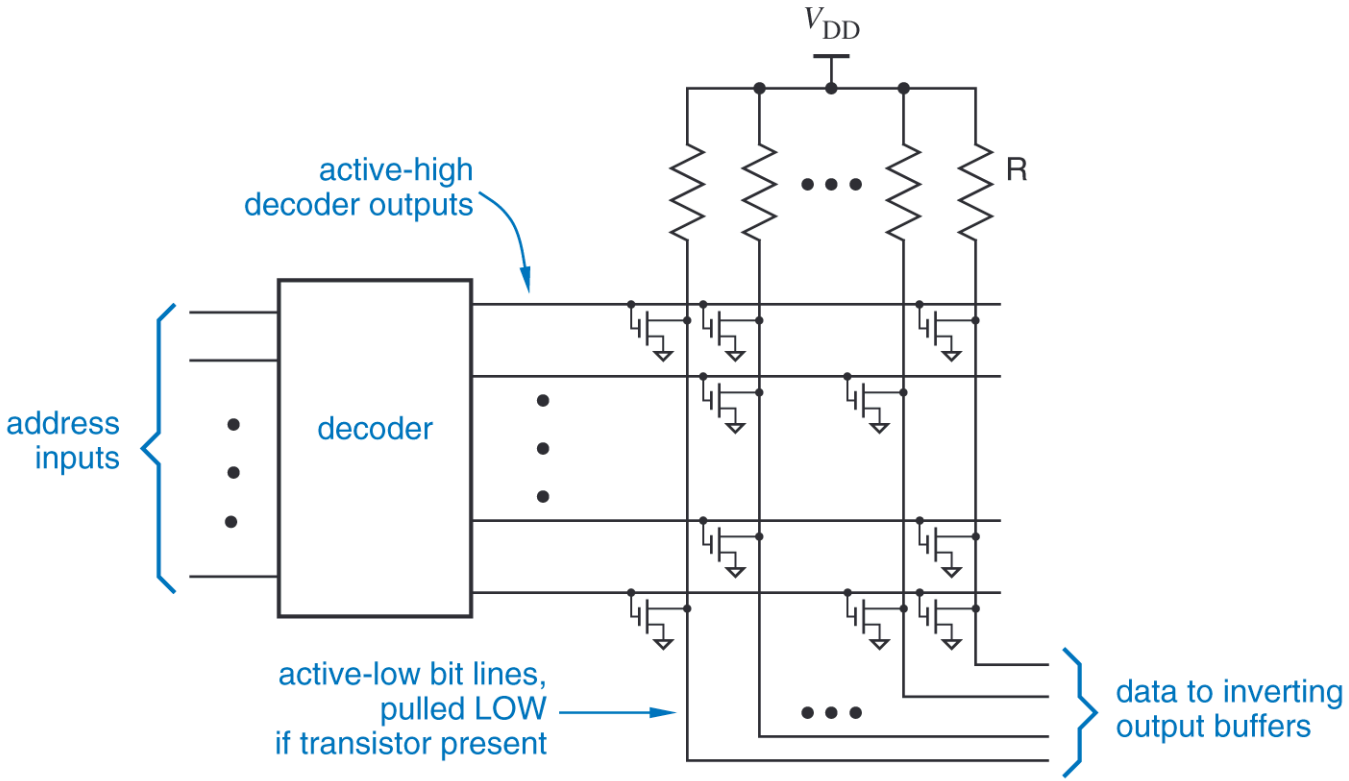


Figure 9-8
MOS transistors as storage elements in a ROM.

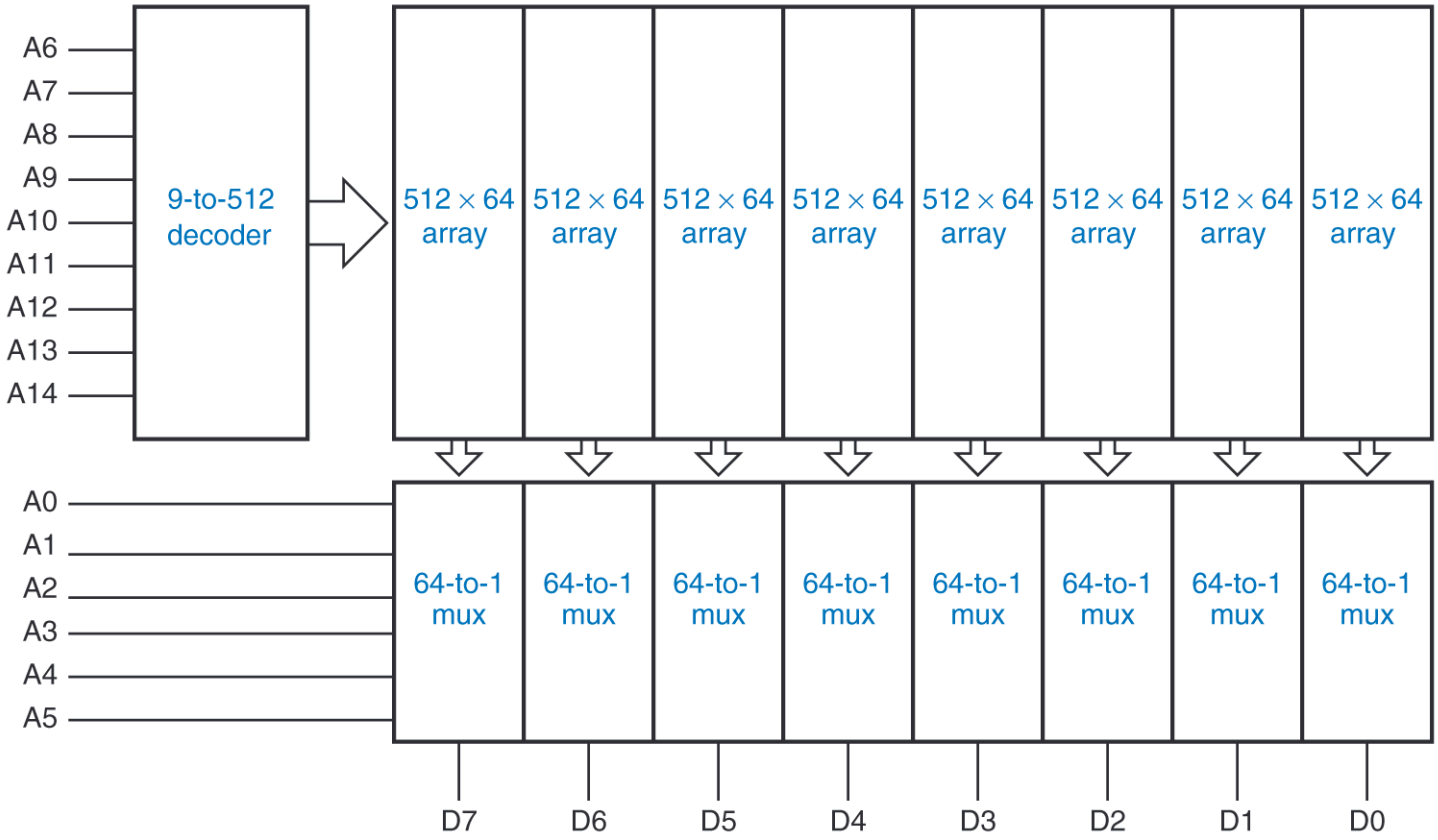


Figure 9-9

Possible layout of a 32K × 8 ROM.

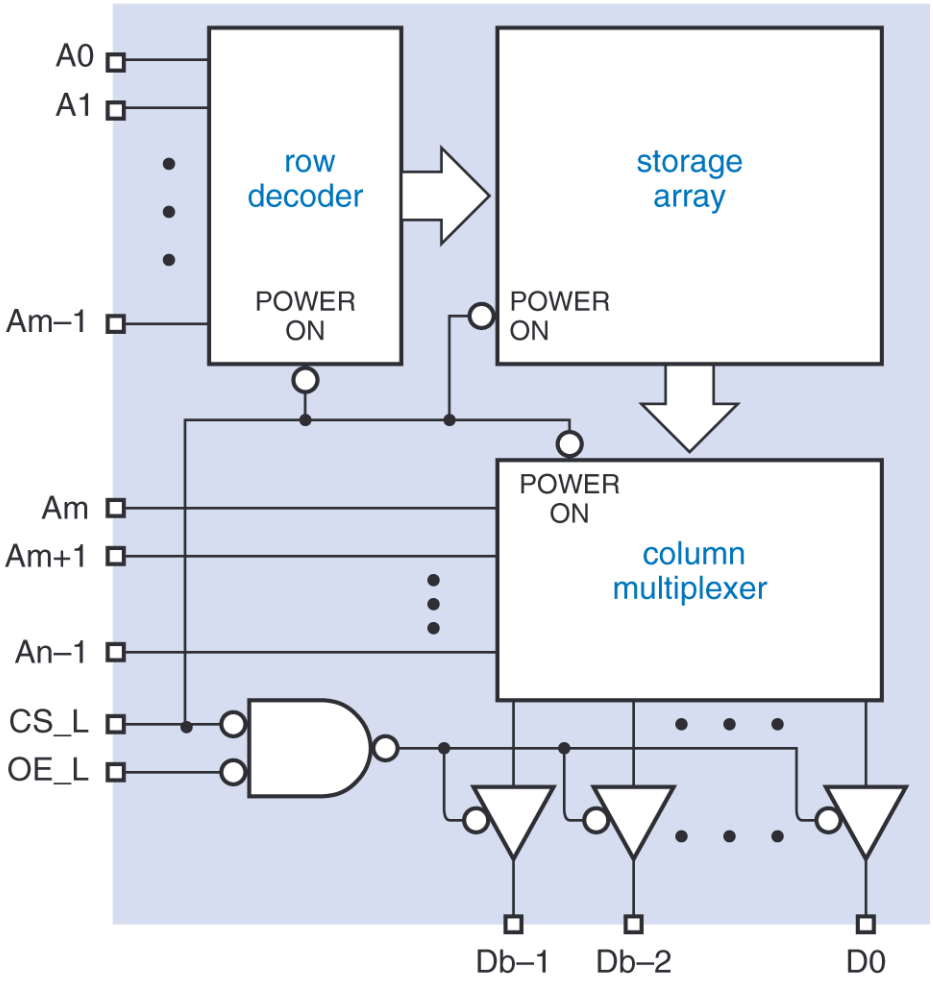


Figure 9-13

Internal ROM structure, showing use of control inputs.

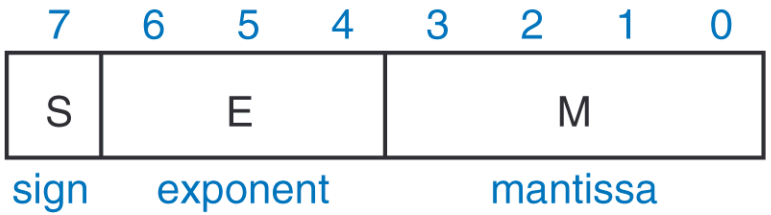


Figure 9-15
Format of a μ-law PCM byte.

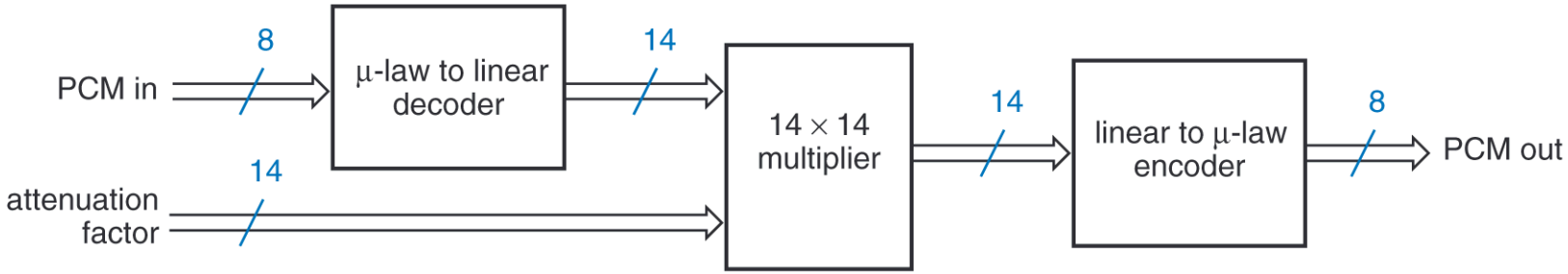


Figure 9-16

Block diagram of a digital attenuator using discrete components.

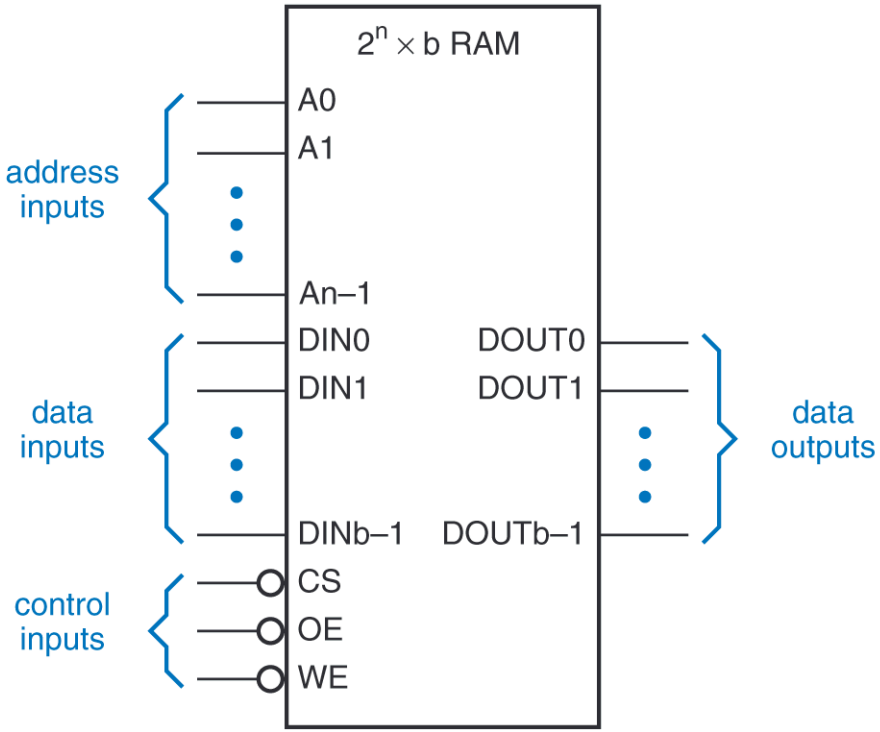


Figure 9-19

Basic structure of a $2^n \times b$ RAM.

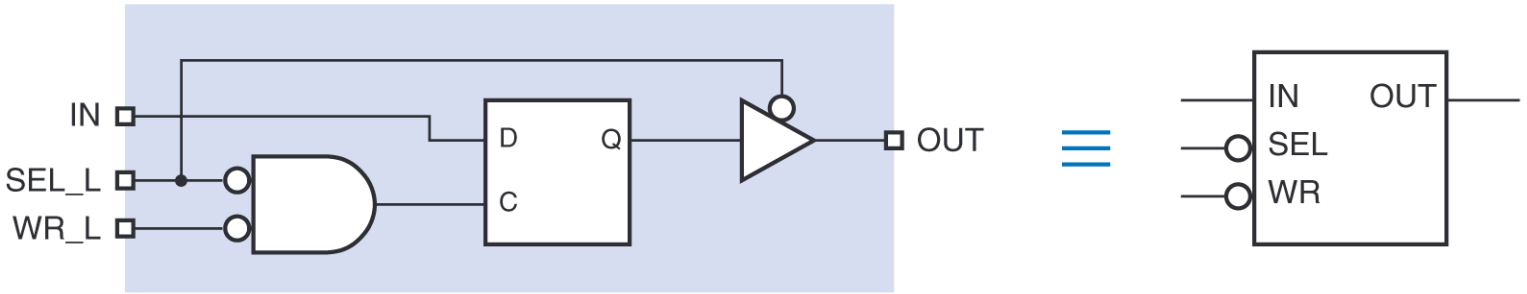


Figure 9-20

Functional behavior of a static-RAM cell.

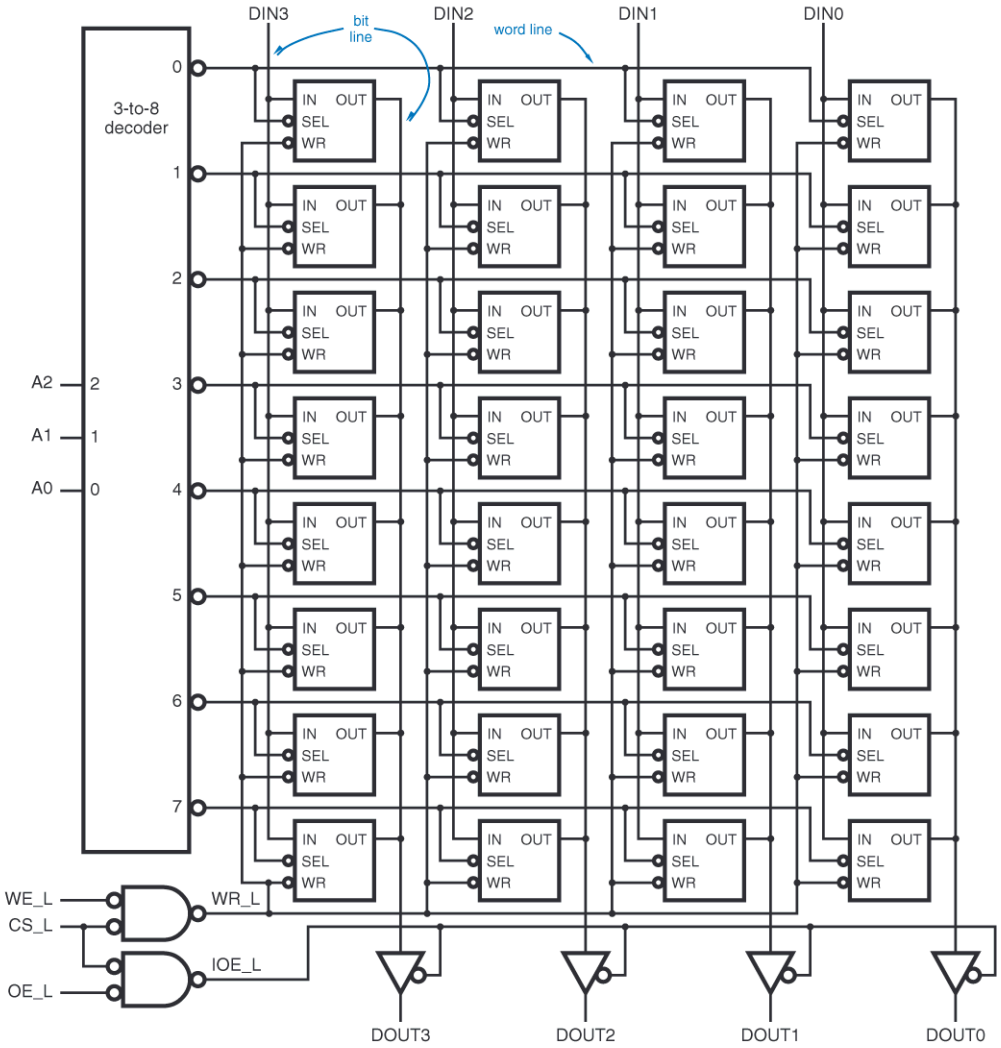


Figure 9-21

Internal structure of an 8 × 4 static RAM.

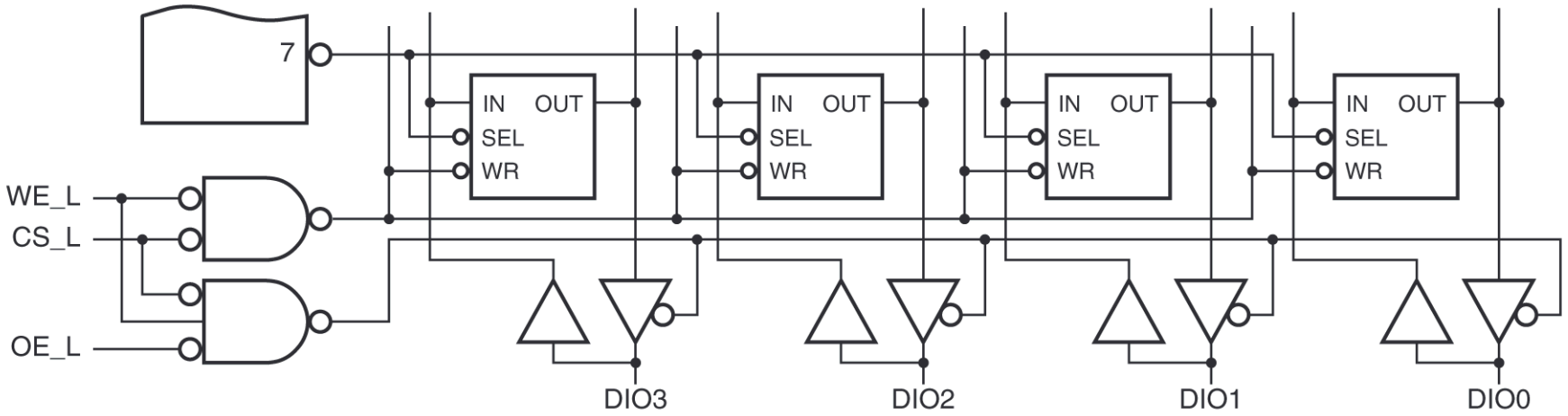


Figure 9-25

Output-buffer control in an SRAM with a bidirectional data bus.

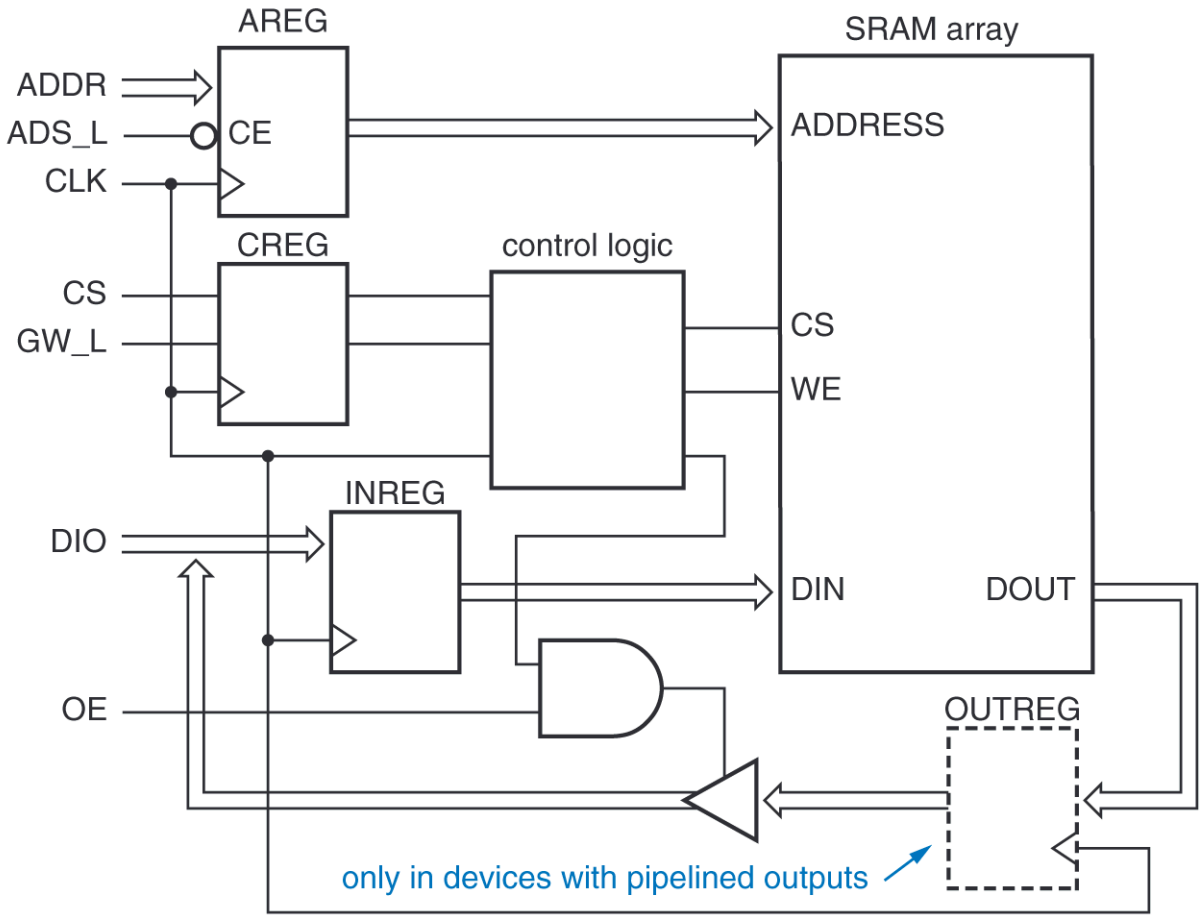


Figure 9-26

Internal structure of a synchronous SRAM.

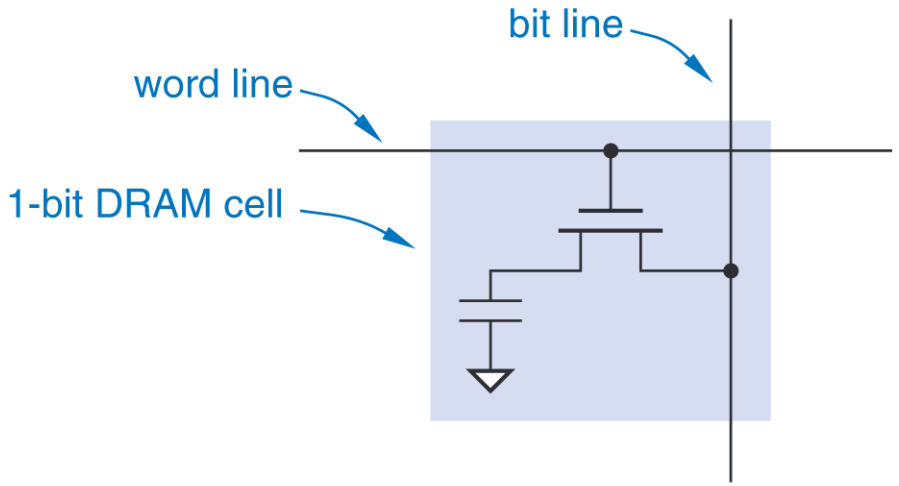


Figure 9-31

Storage cell for one bit in a DRAM.

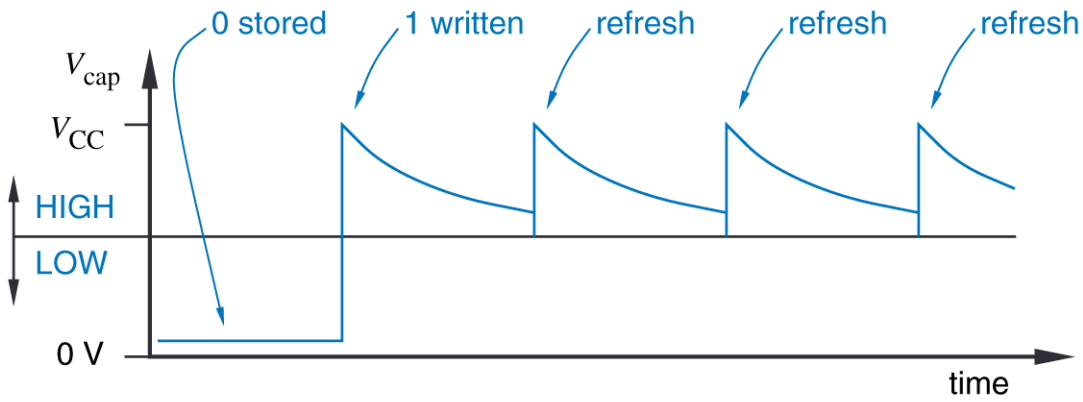


Figure 9-32

Voltage stored in a DRAM cell after writing and refresh operations.

Command Name	*Description*
NOP	No operation
ACTV	Row-address strobe and activate bank
READ	Column address and read command
READA	Read with auto-precharge
WRIT	Column address and read command
WRITA	Write with auto-precharge
REF	Auto refresh
PRE	Precharge

Table 9-8

Commonly used SDRAM commands.

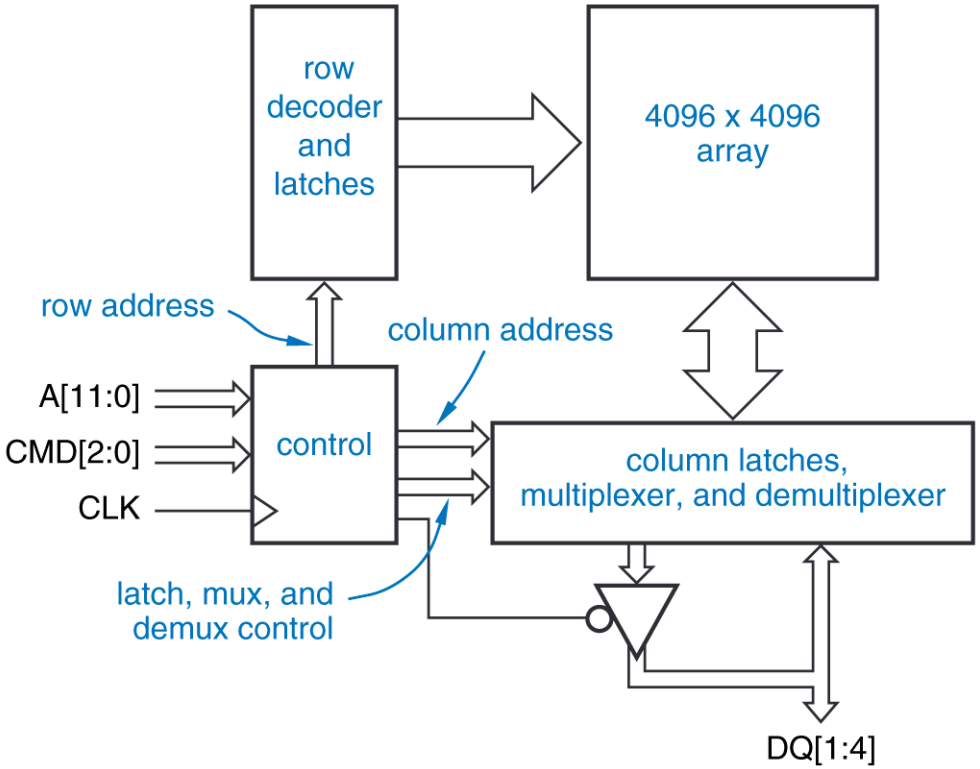


Figure 9-33

SDRAM internal structure.

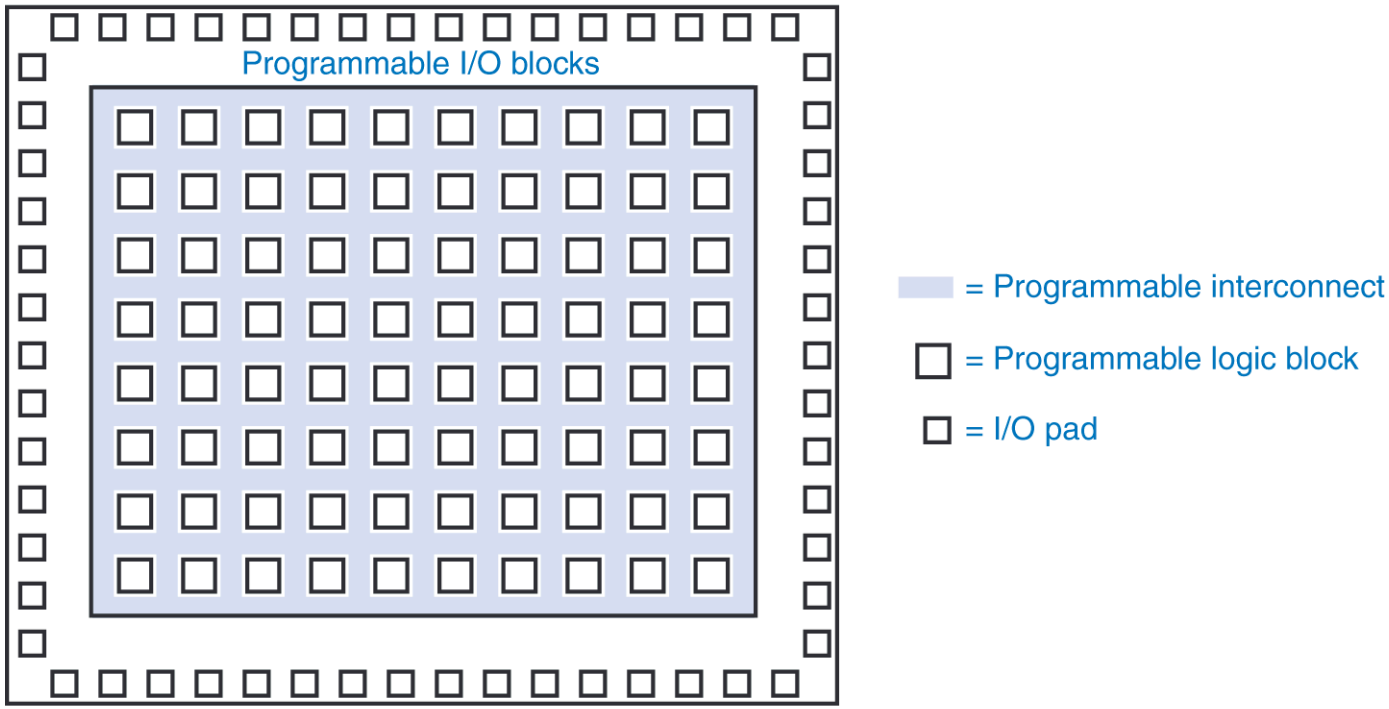


Figure 9-44

General FPGA chip architecture.

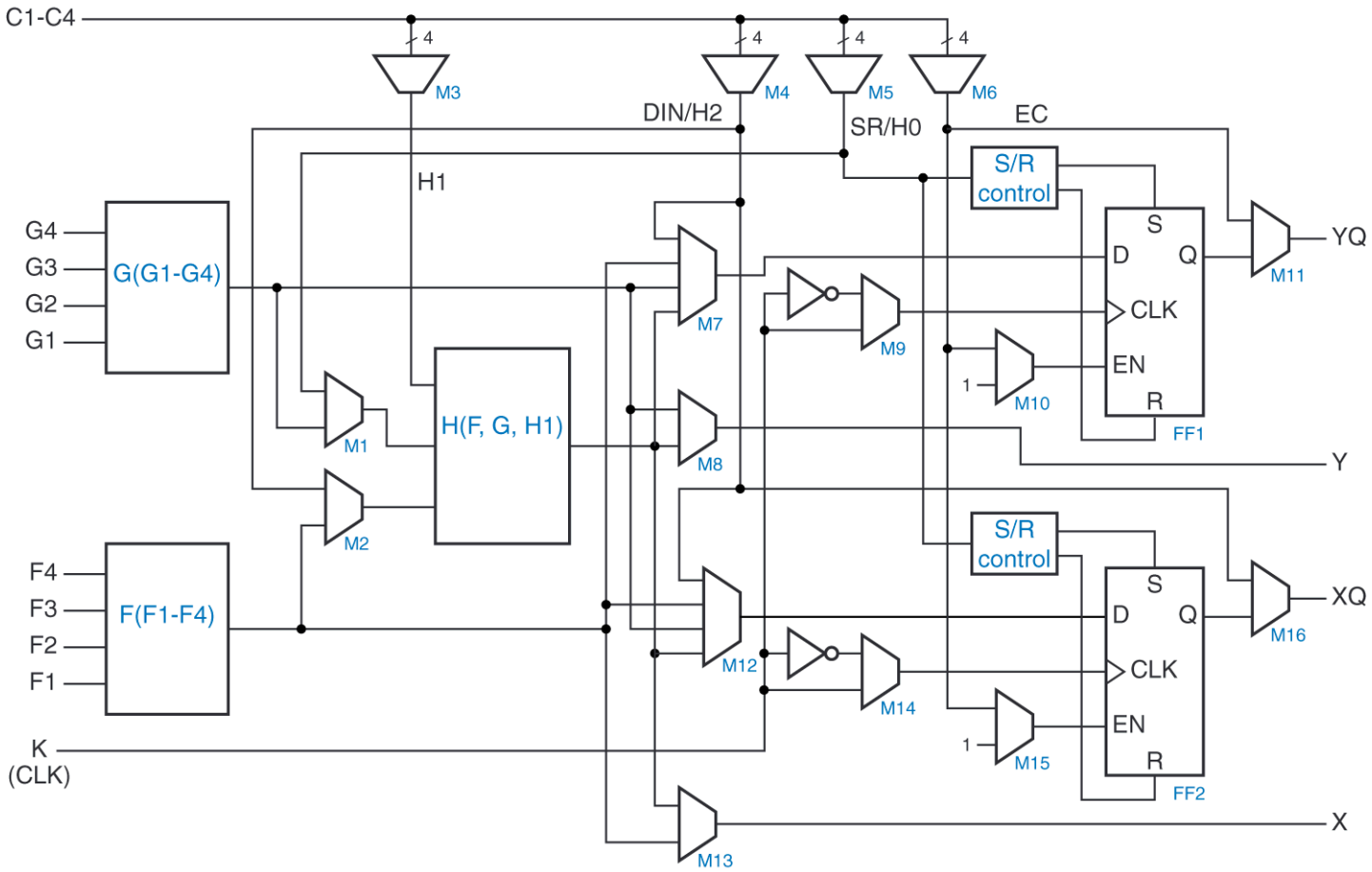


Figure 9-45

XC4000 configurable logic block.

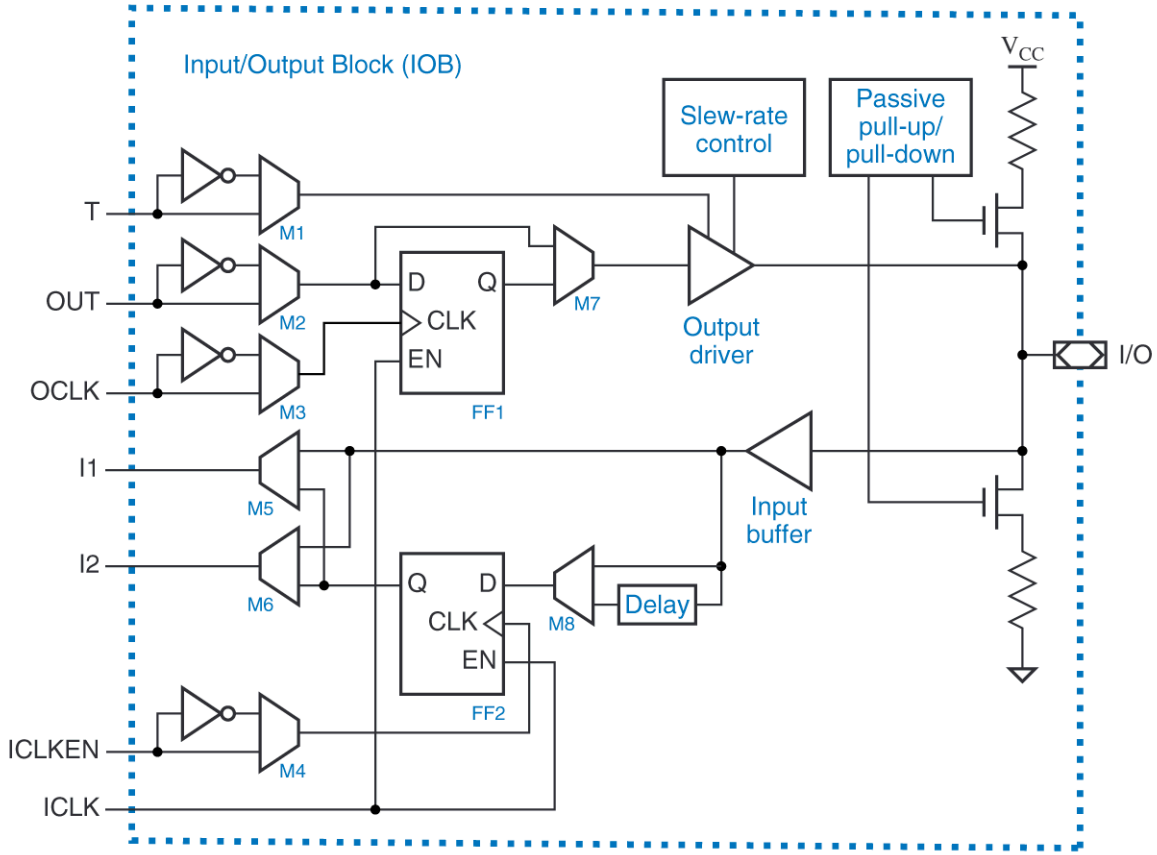


Figure 9-46

XC4000 I/O block.

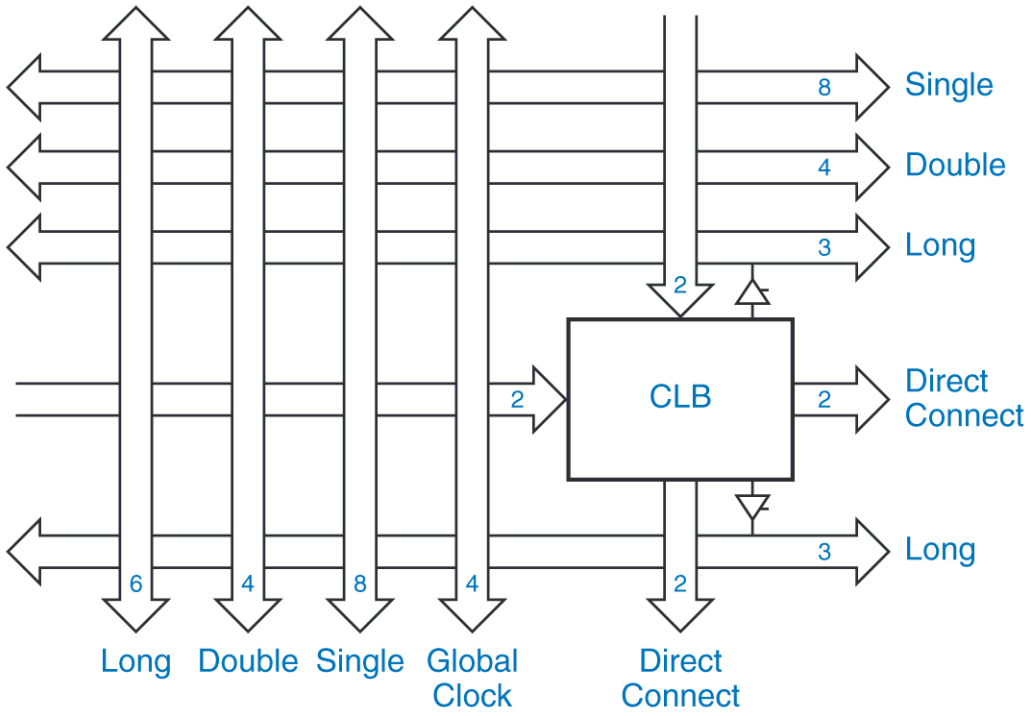


Figure 9-47

XC4000 general interconnect structure.

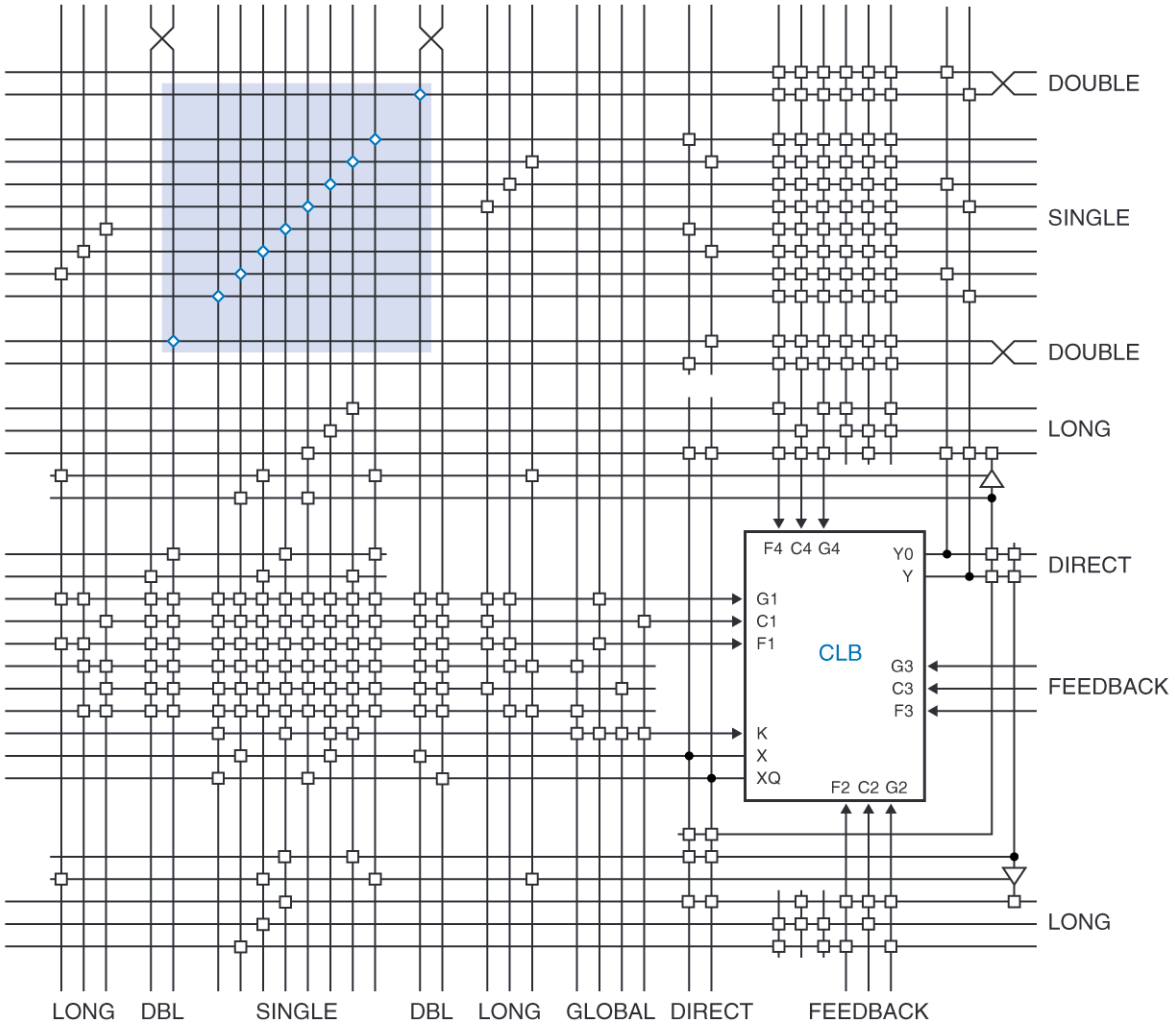


Figure 9-48

XC4000 CLB and wire connection details.

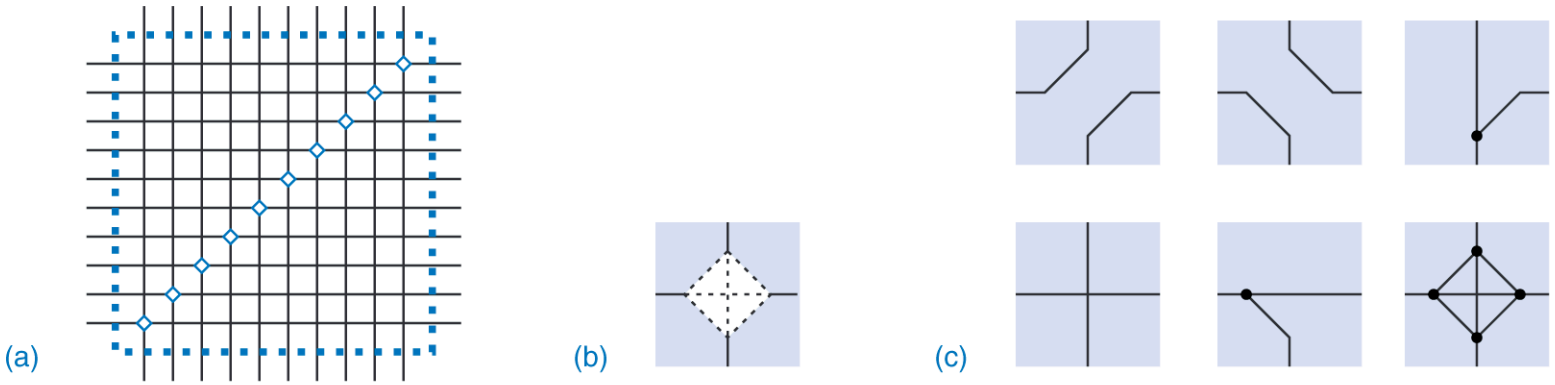
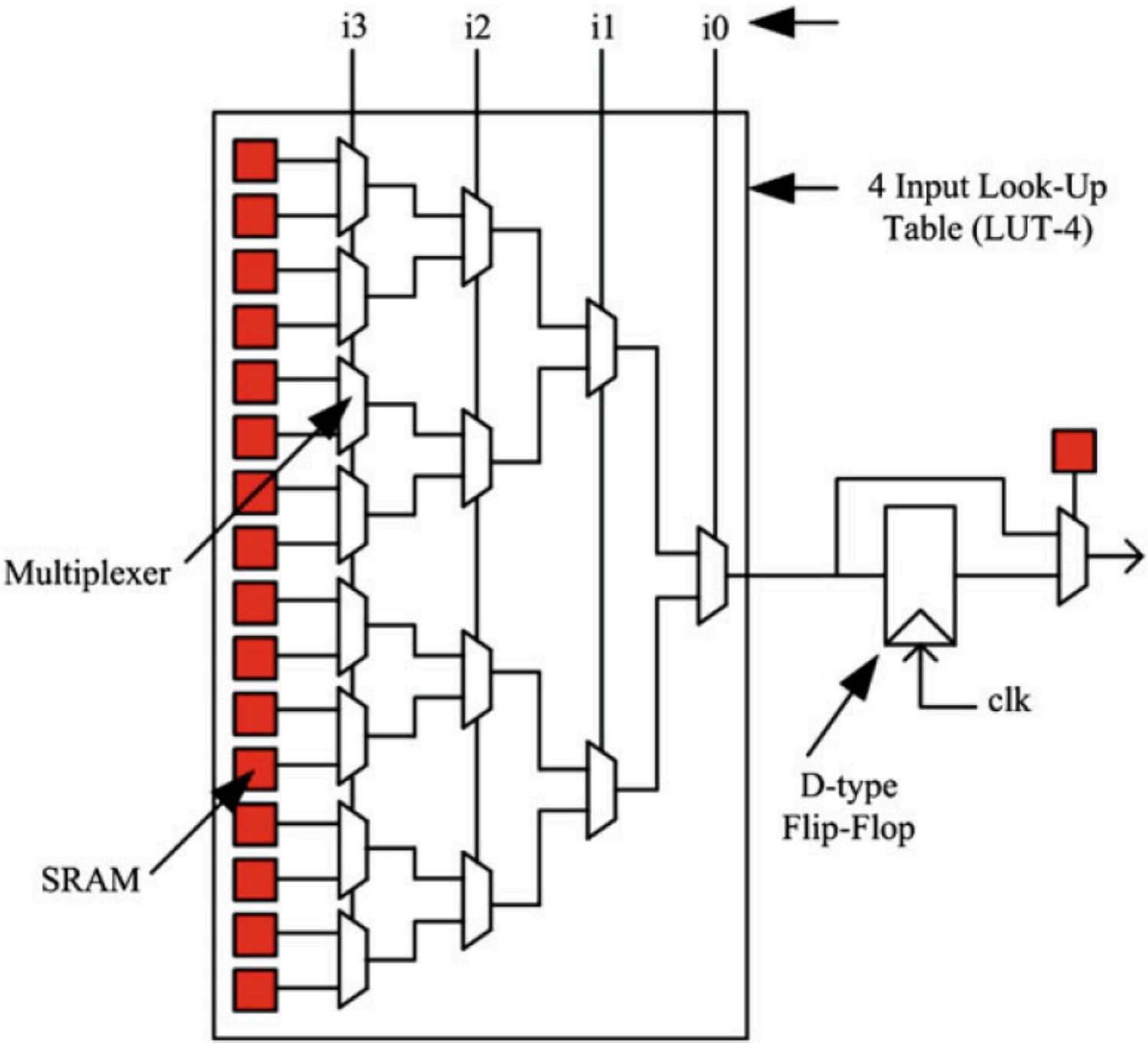
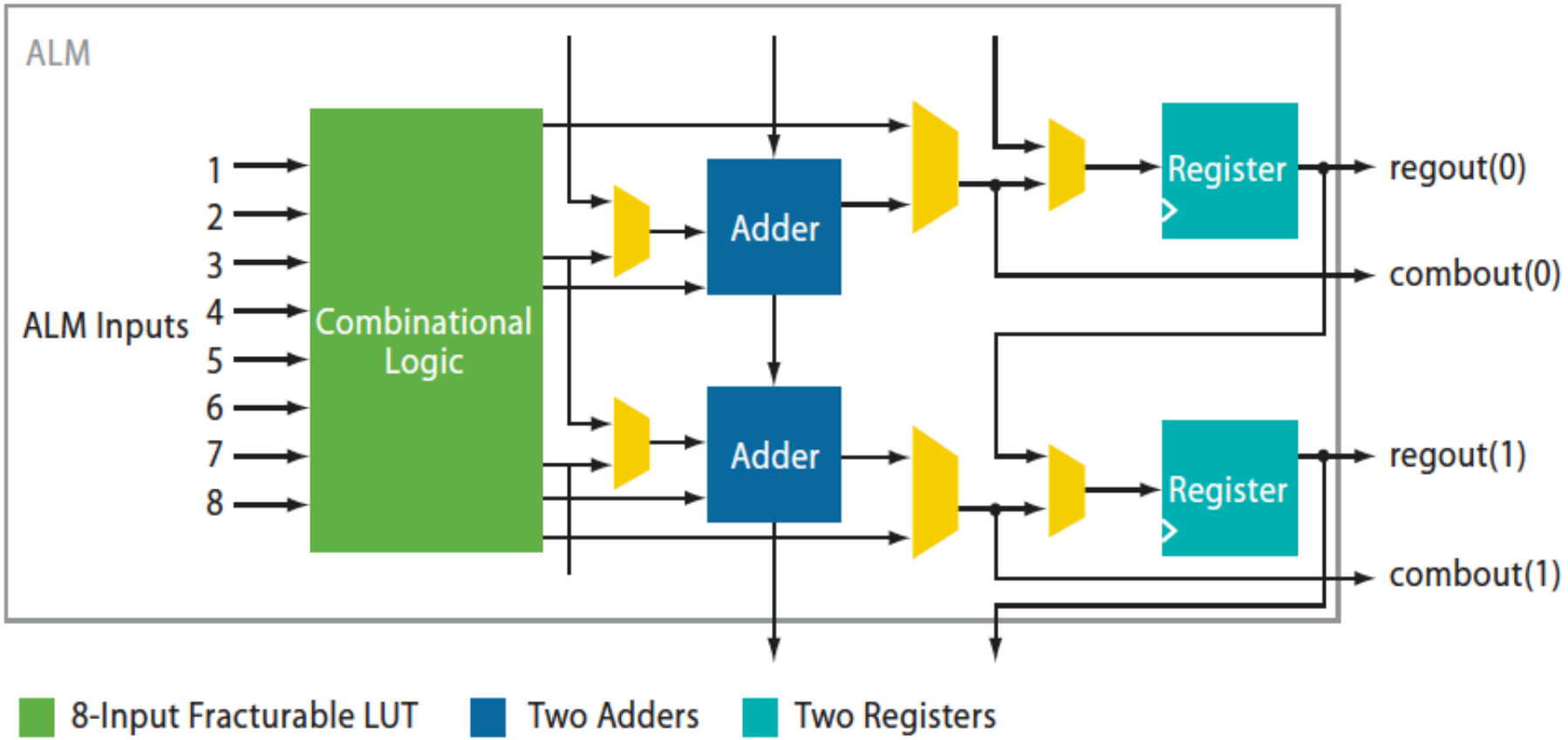
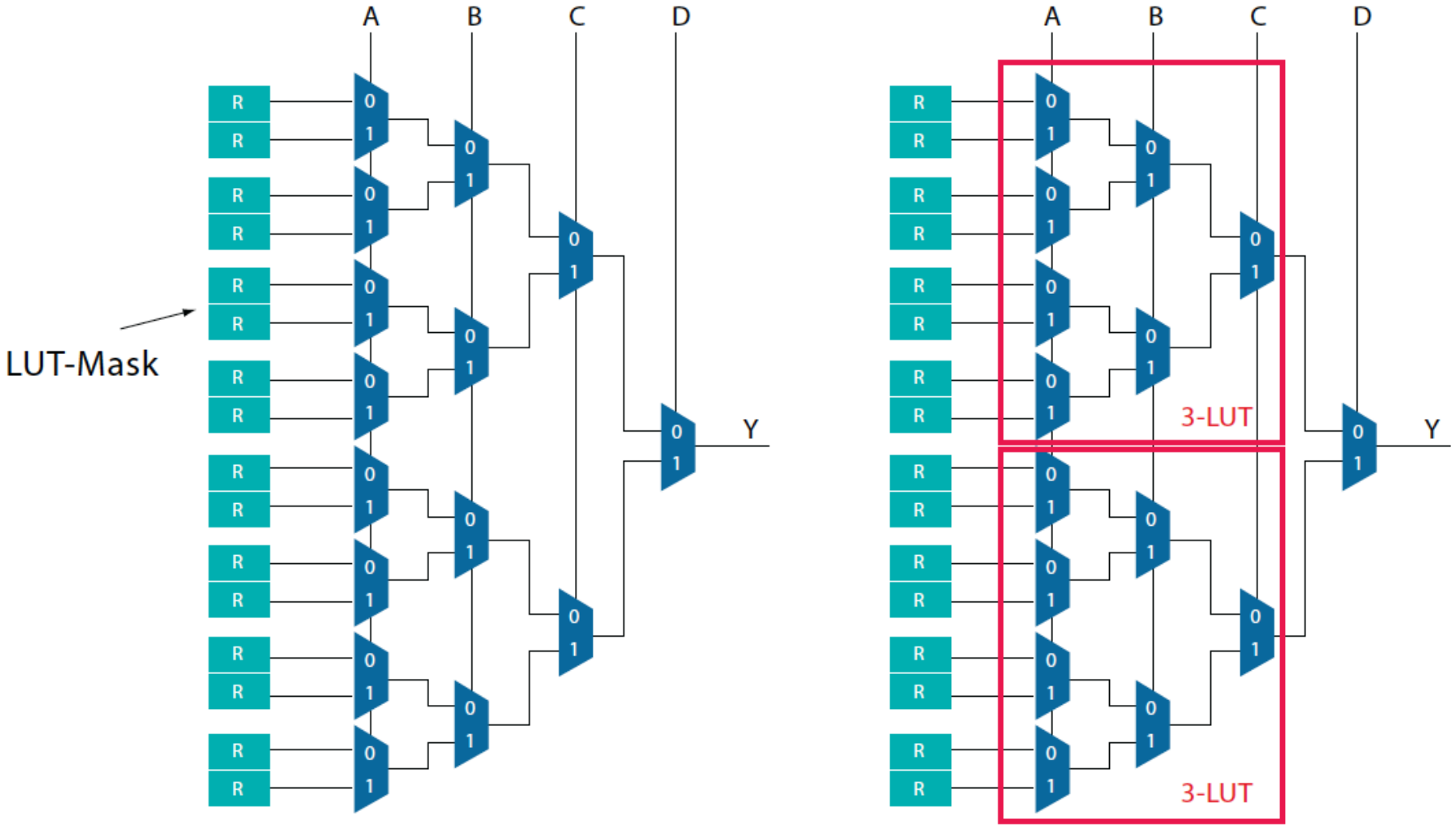


Figure 9-49

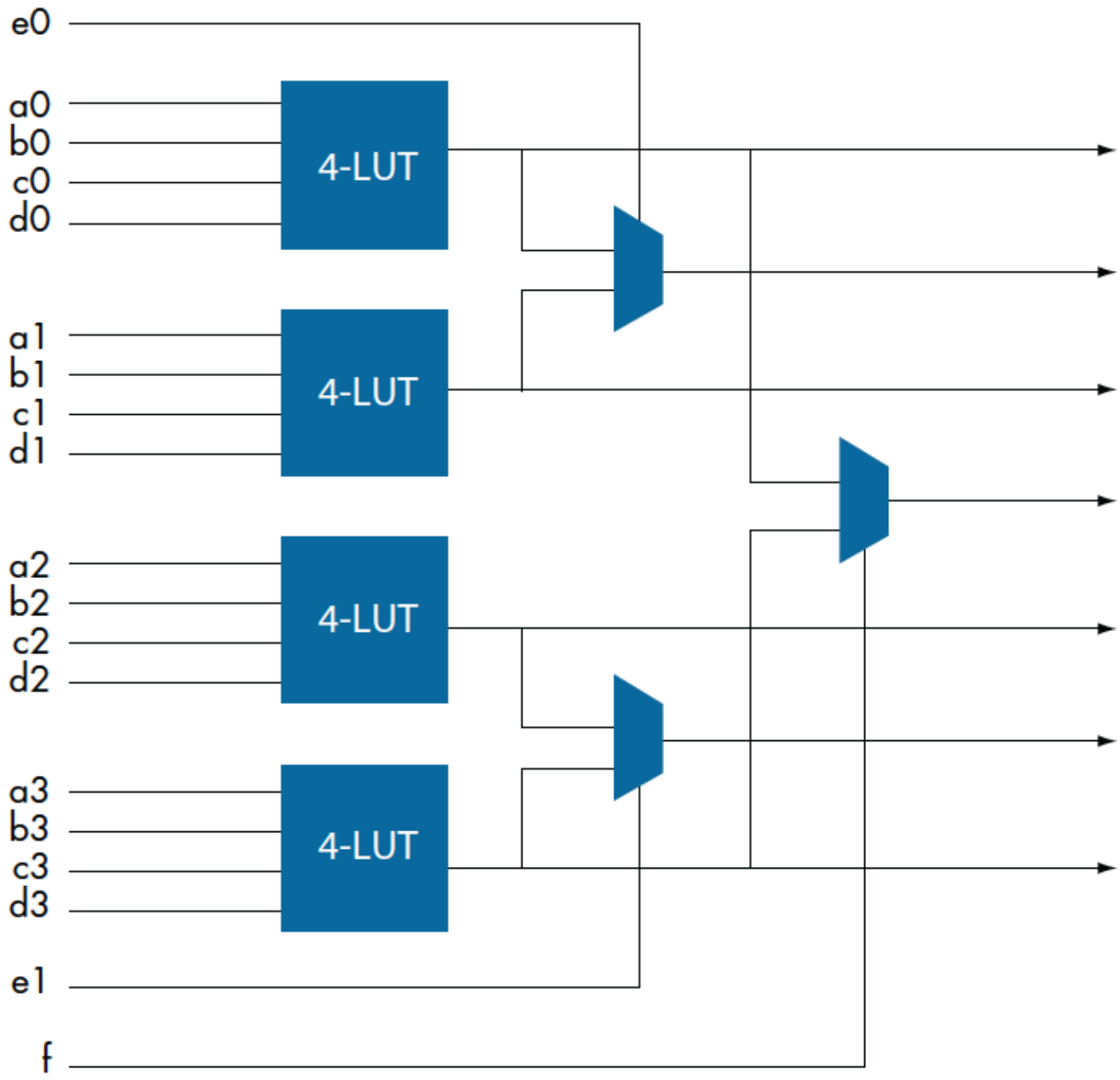
XC4000 programmable connections: (a) programmable switch matrix (PSM); (b) programmable switch element (PSE); (c) a few possible connections.

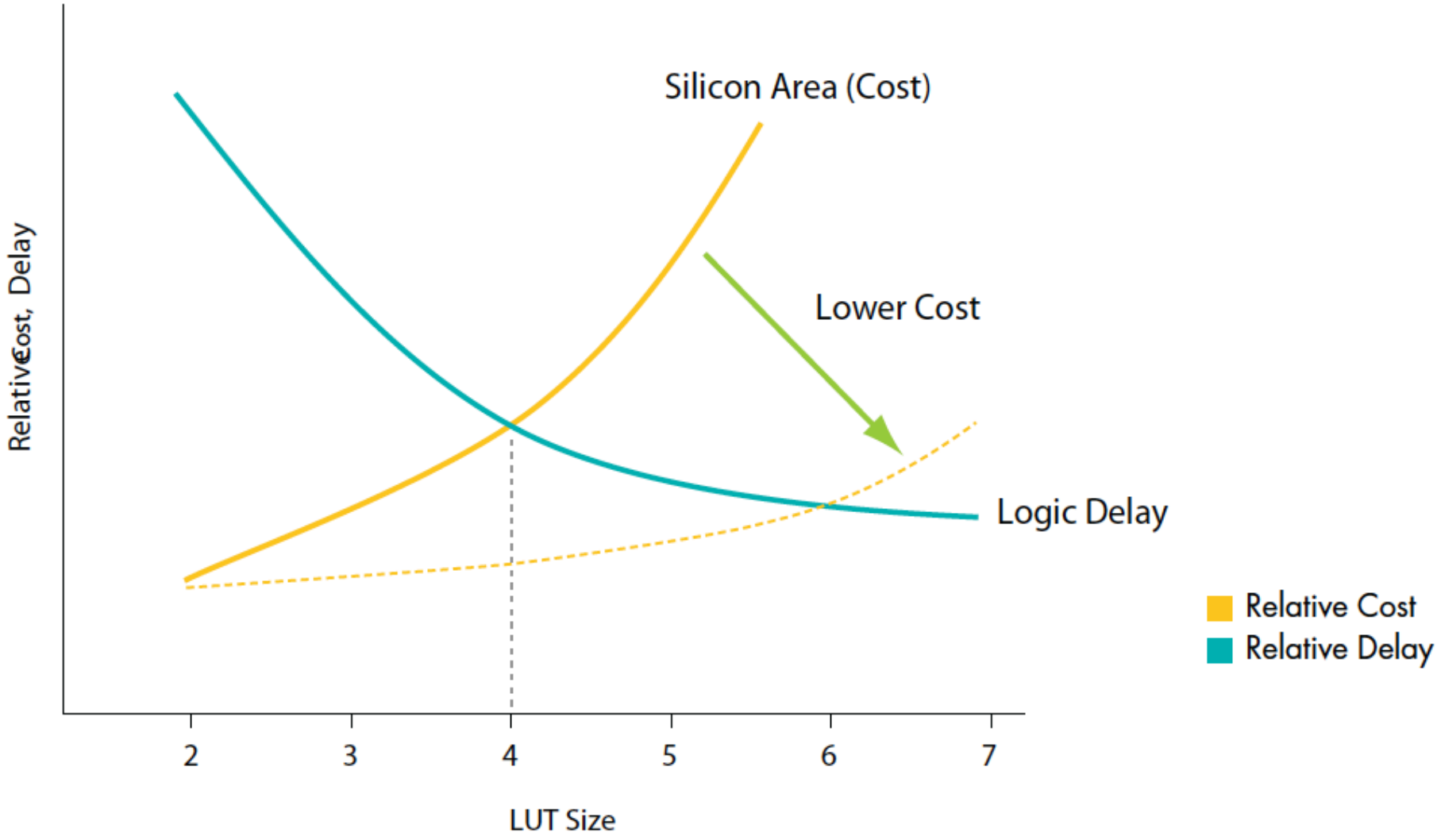

i3
i2
i1
i0
4 Input Look-Up
Table (LUT-4)
Multiplexer
SRAM
D-type
Flip-Flop
clk

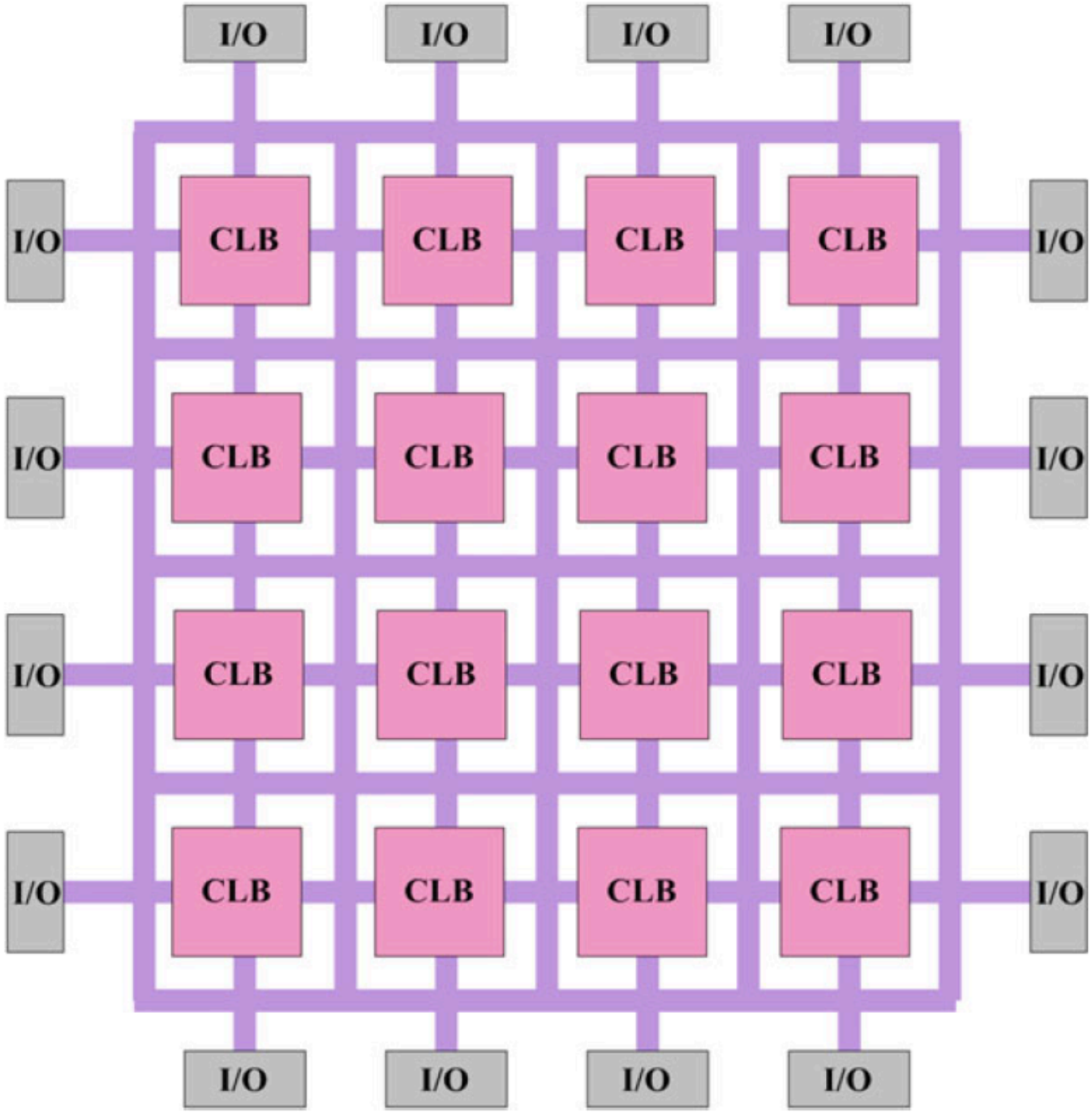

ALM
ALM Inputs
1
2
3
4
5
6
7
8
Combinational Logic
Adder
Adder
Register
Register
regout(0)
combout(0)
regout(1)
combout(1)
8-Input Fracturable LUT
Two Adders
Two Registers



a'b'c'd' + abcd + abc'd' = 1000 0000 0000 1001 = 0x8009


e0
a0
b0
c0
d0
4-LUT
a1
b1
c1
d1
4-LUT
a2
b2
c2
d2
4-LUT
a3
b3
c3
d3
4-LUT
e1
f


Silicon Area (Cost)
Lower Cost
Logic Delay
Relative Cost, Delay
2
3
4
5
6
7
LUT Size
Relative Cost
Relative Delay


I/O
I/O
I/O
I/O
I/O
CLB
CLB
CLB
CLB
I/O
I/O
CLB
CLB
CLB
CLB
I/O
I/O
CLB
CLB
CLB
CLB
I/O
I/O
CLB
CLB
CLB
CLB
I/O
I/O
I/O
I/O
I/O