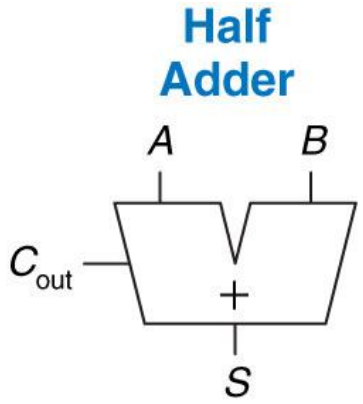
Chapter 5

Digital Building Blocks



A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = A \oplus B$$

$$C_{out} = AB$$

Figure 5.1 1-bit half adder

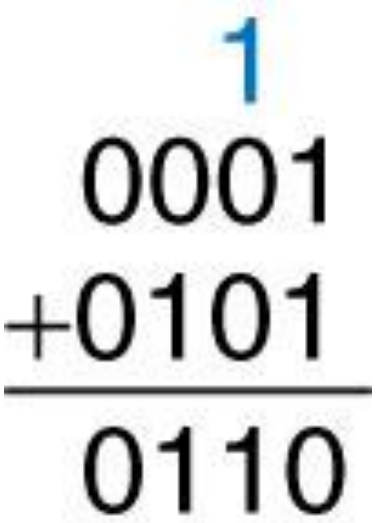
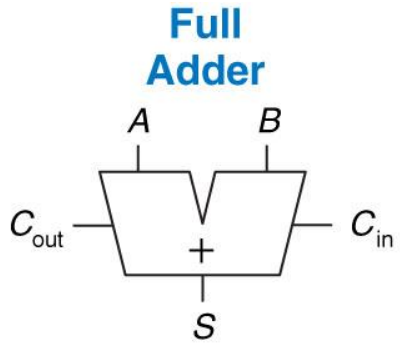


Figure 5.2 Carry bit



C_{in}	A	B	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + AC_{in} + BC_{in}$$

Figure 5.3 1-bit full adder

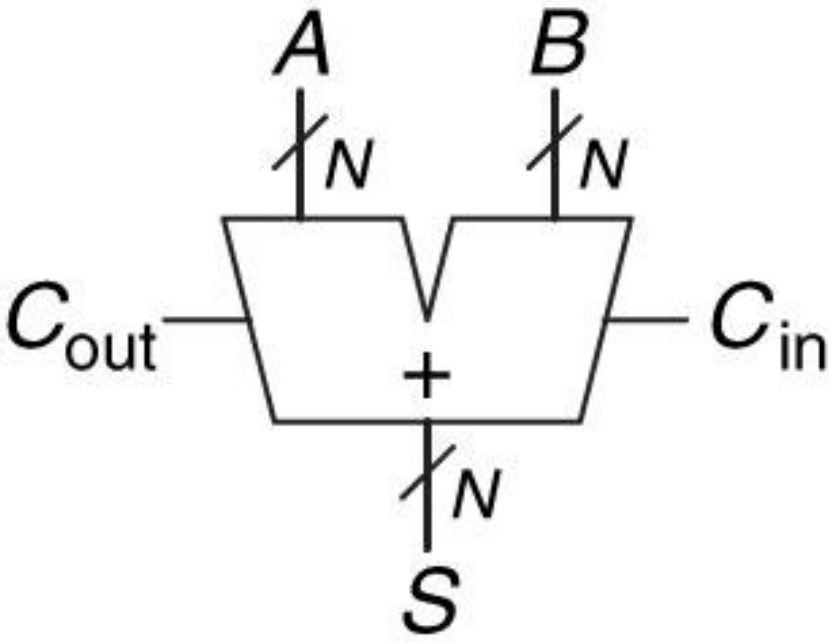


Figure 5.4 Carry propagate adder

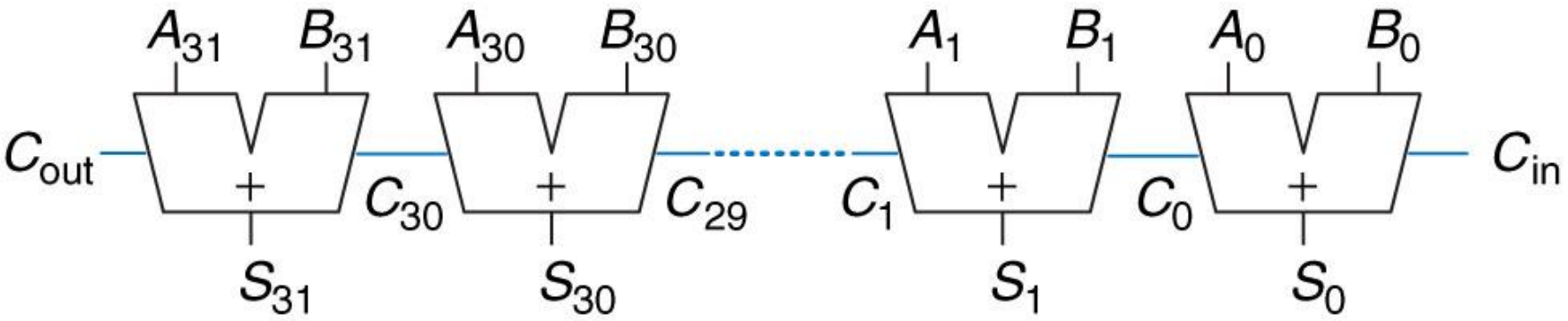


Figure 5.5 32-bit ripple-carry adder

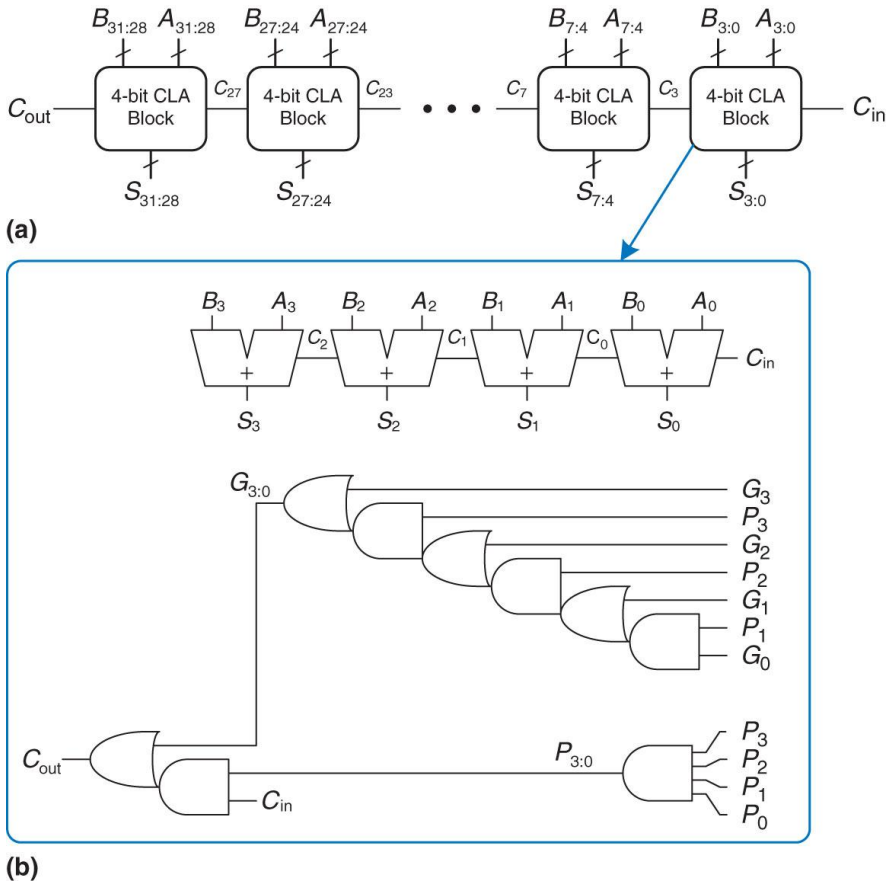


Figure 5.6 (a) 32-bit carry-lookahead adder (CLA), (b) 4-bit CLA block

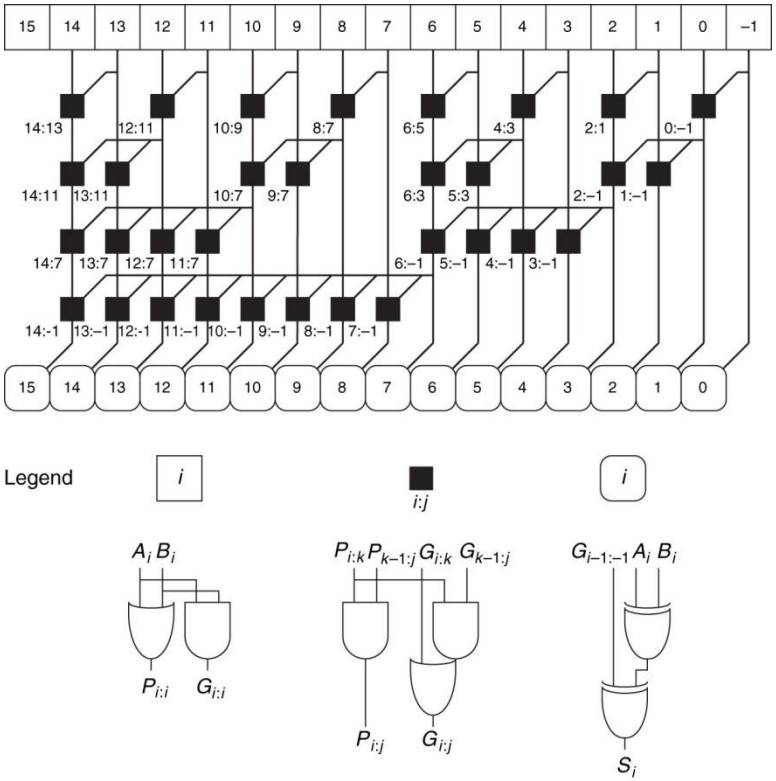


Figure 5.7 16-bit prefix adder

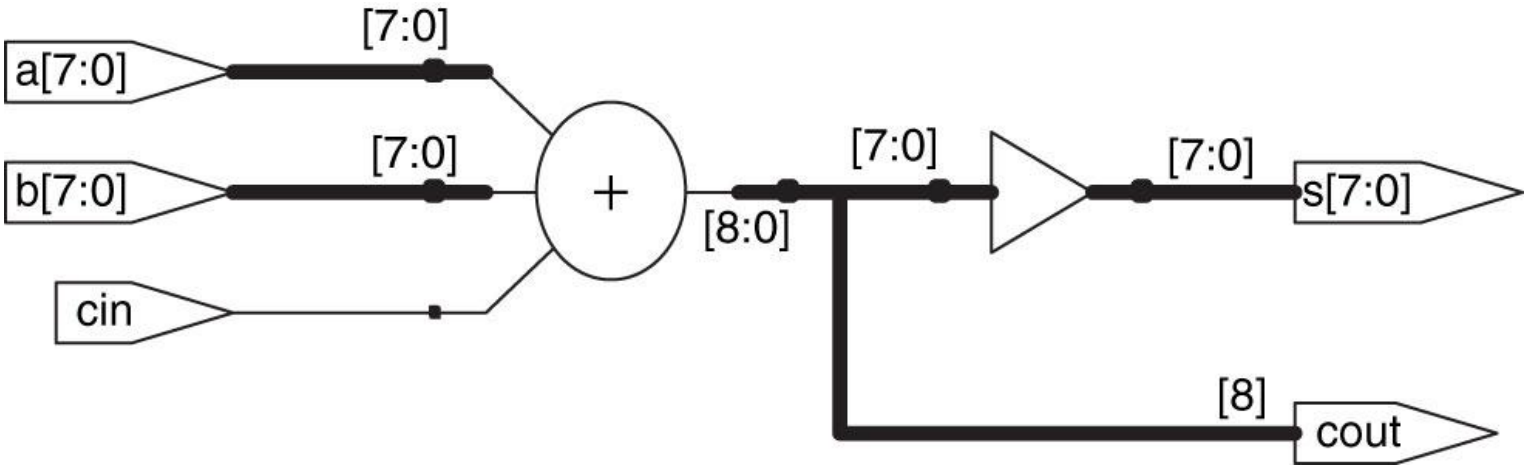


Figure 5.8 Synthesized adder

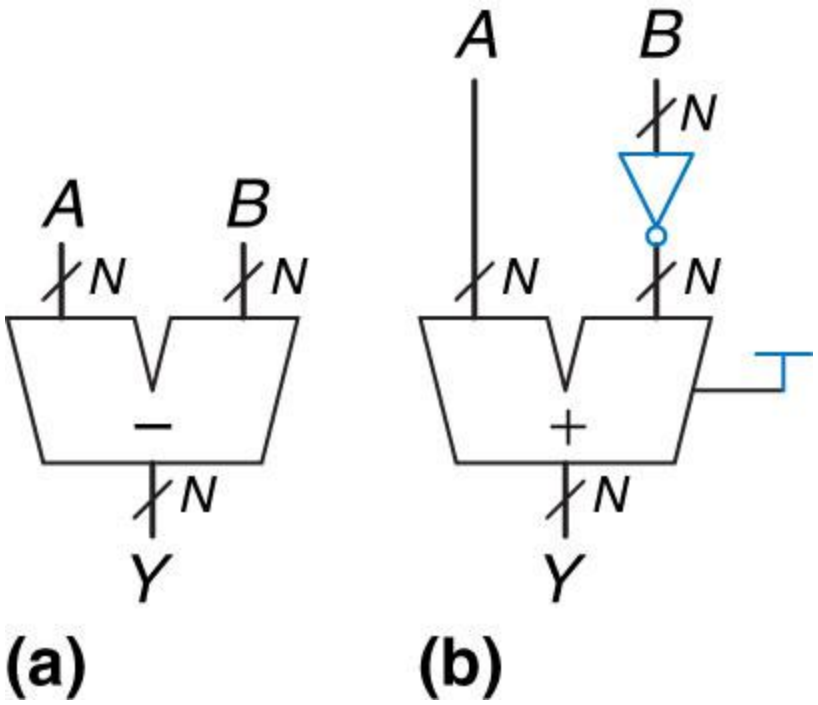


Figure 5.9 Subtractor: (a) symbol, (b) implementation

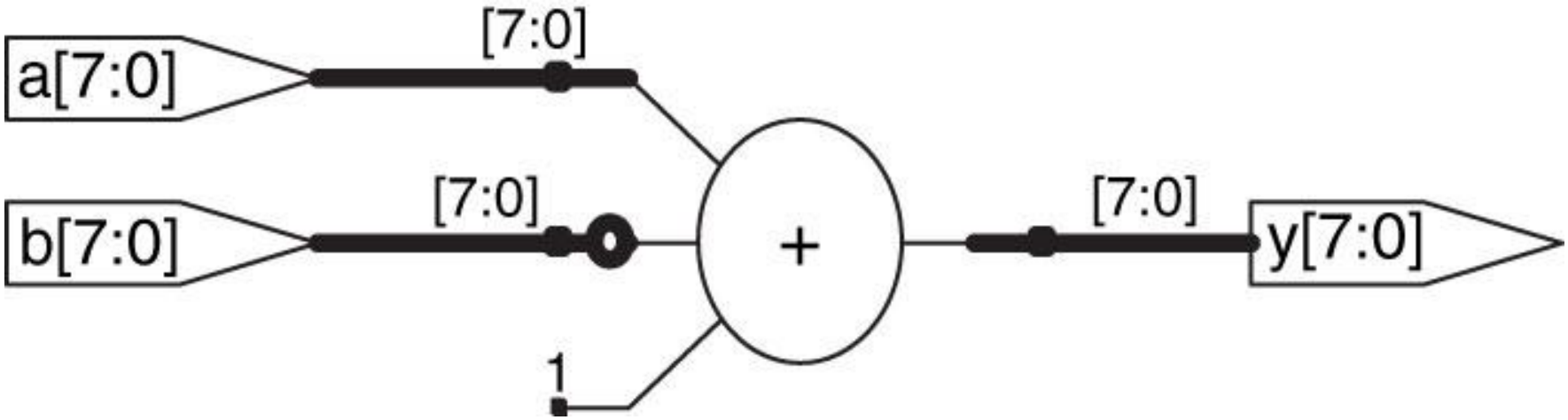


Figure 5.10 Synthesized subtractor

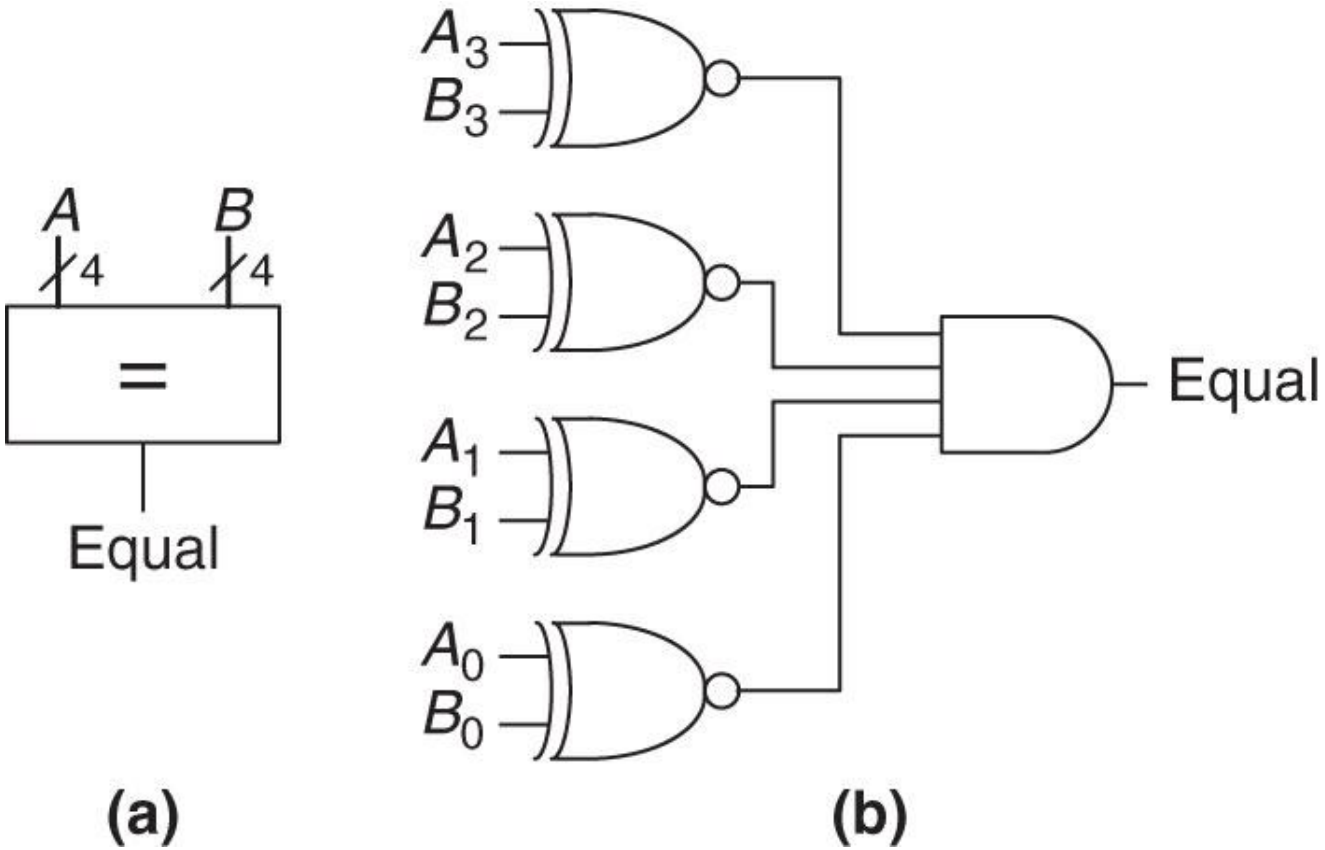


Figure 5.11 4-bit equality comparator: (a) symbol, (b) implementation

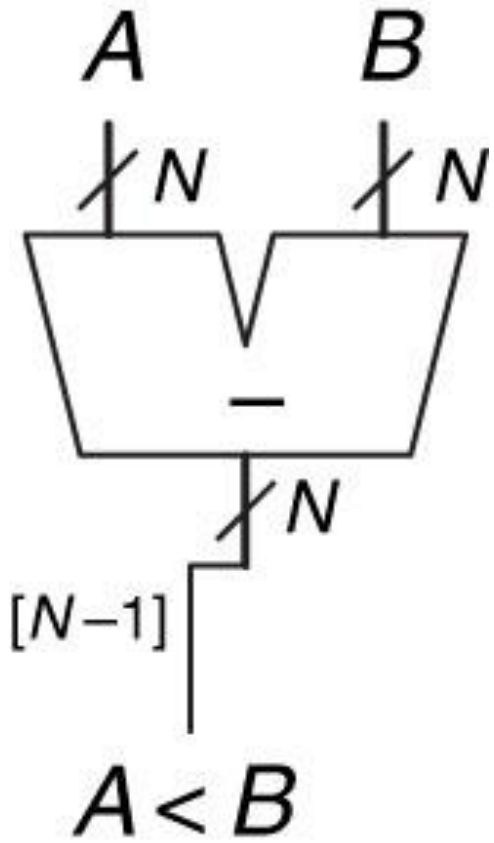


Figure 5.12 *N*-bit magnitude comparator

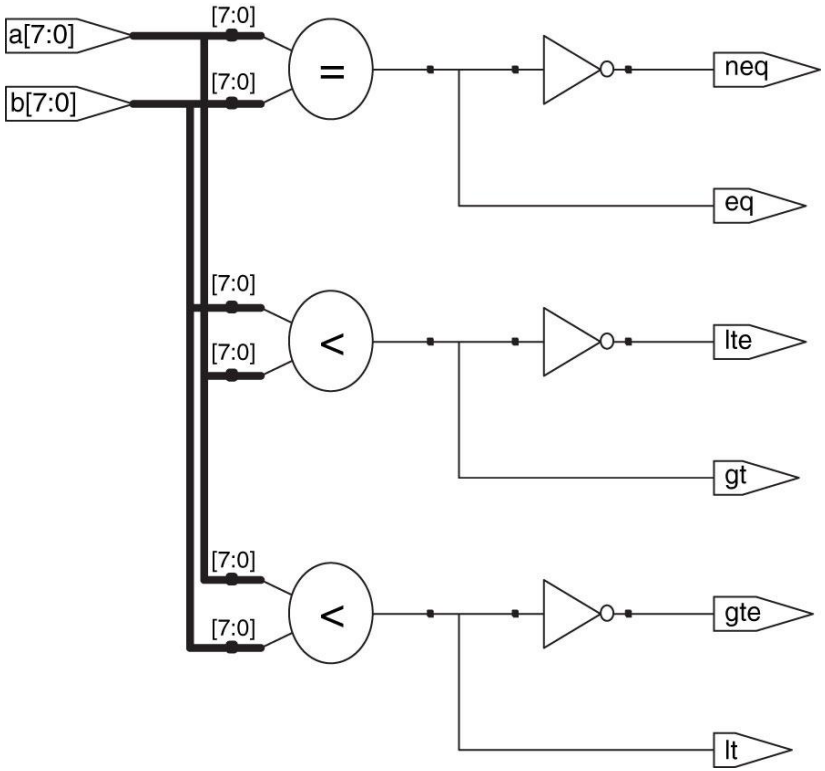


Figure 5.13 Synthesized comparators

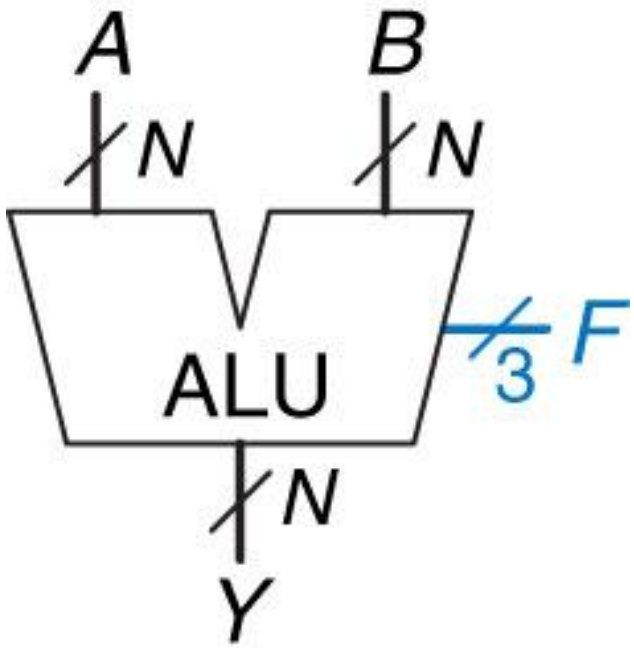


Figure 5.14 ALU symbol

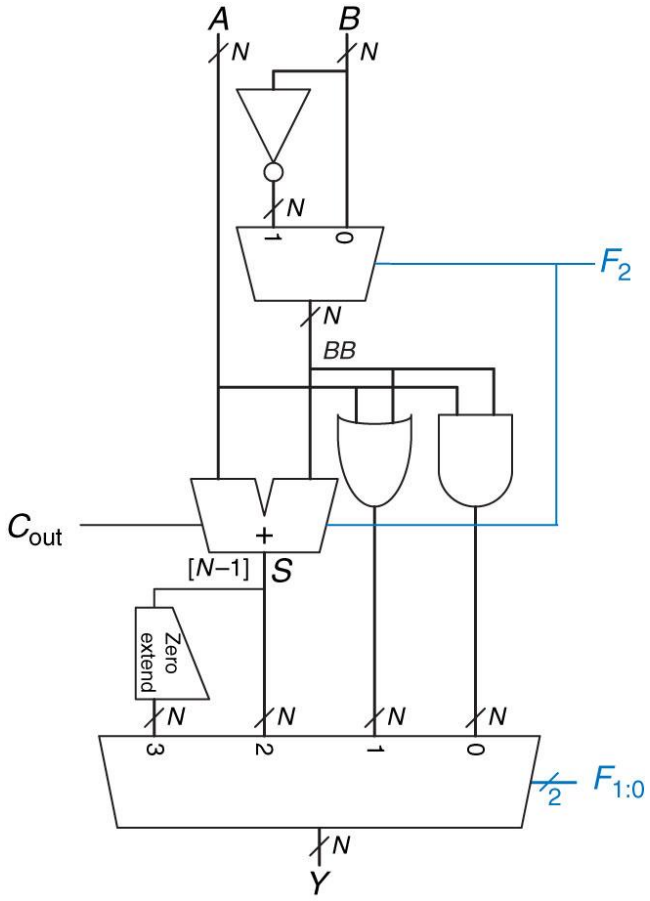


Figure 5.15 *N*-bit ALU

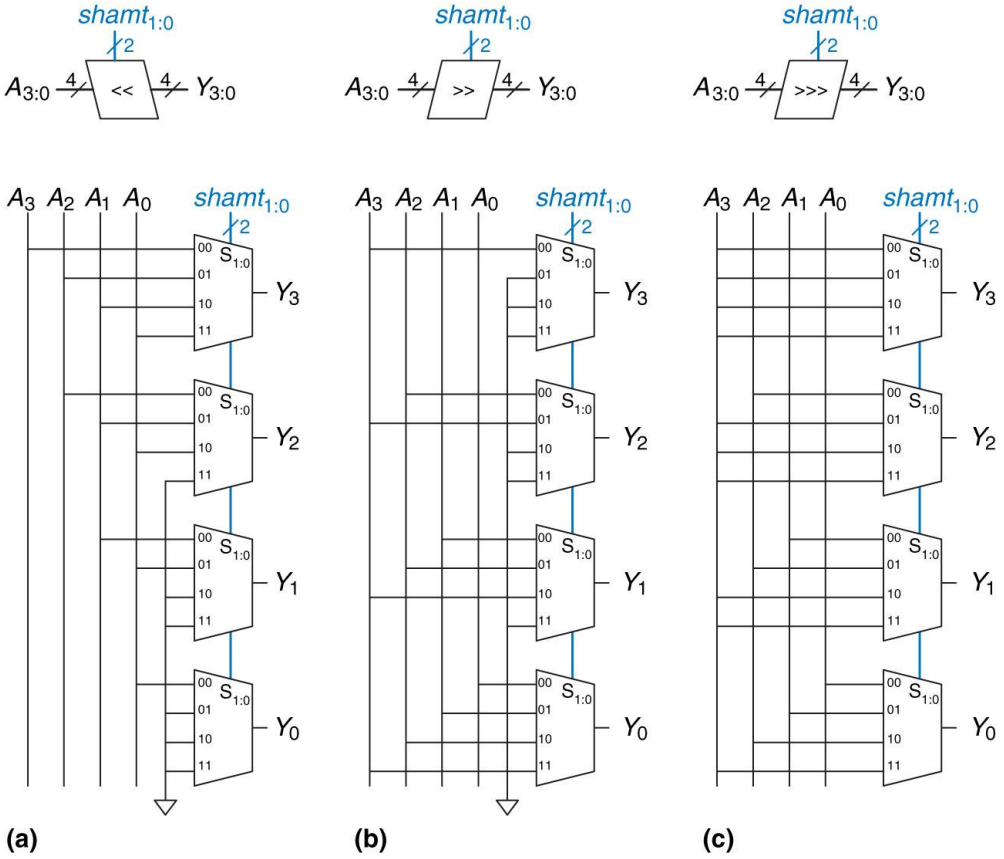


Figure 5.16 4-bit shifters: (a) shift left, (b) logical shift right, (c) arithmetic shift right

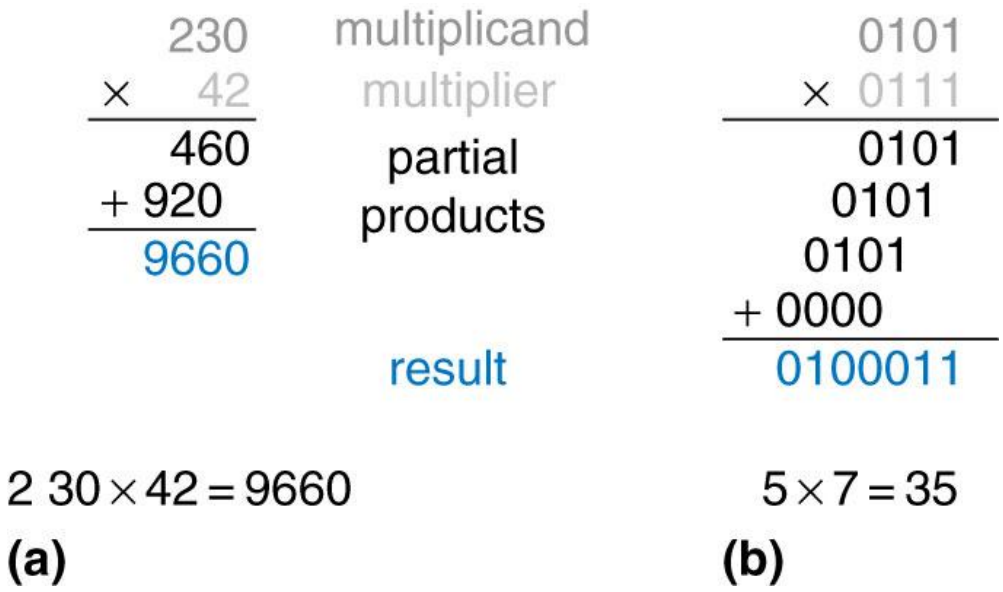


Figure 5.17 Multiplication: (a) decimal, (b) binary

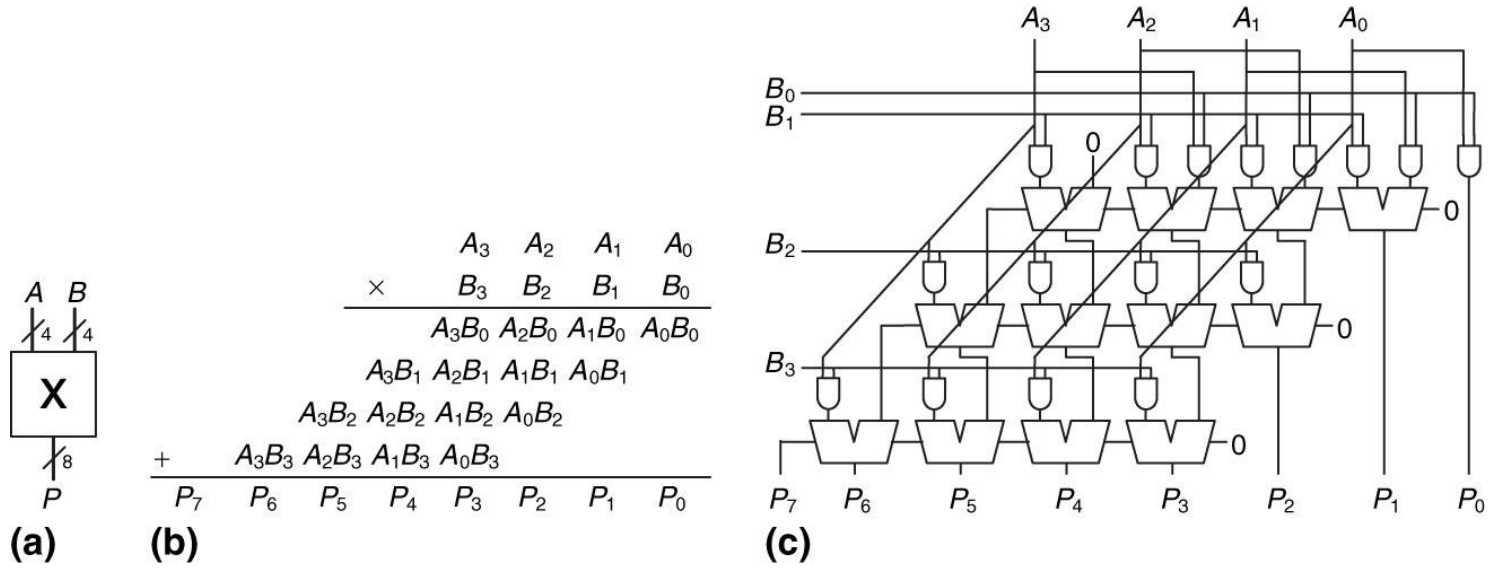


Figure 5.18 4 · 4 multiplier: (a) symbol, (b) function, (c) implementation

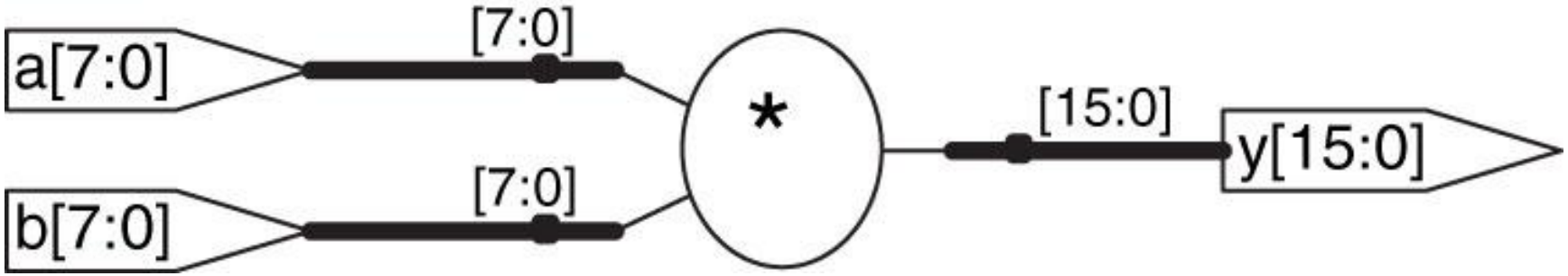


Figure 5.19 Synthesized multiplier

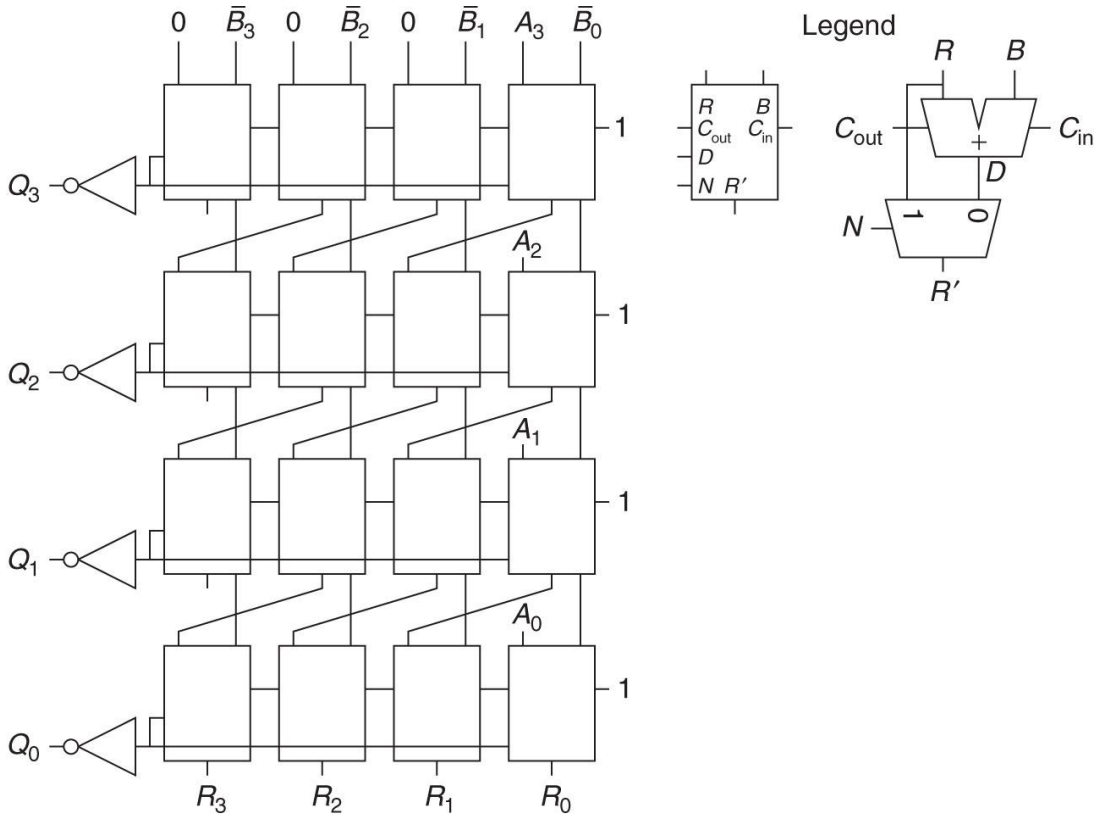


Figure 5.20 Array divider

(a) 01101100

(b) 0110.1100

(c) $2^2 + 2^1 + 2^{-1} + 2^{-2} = 6.75$

Figure 5.21 Fixed-point notation of 6.75 with four integer bits and four fraction bits

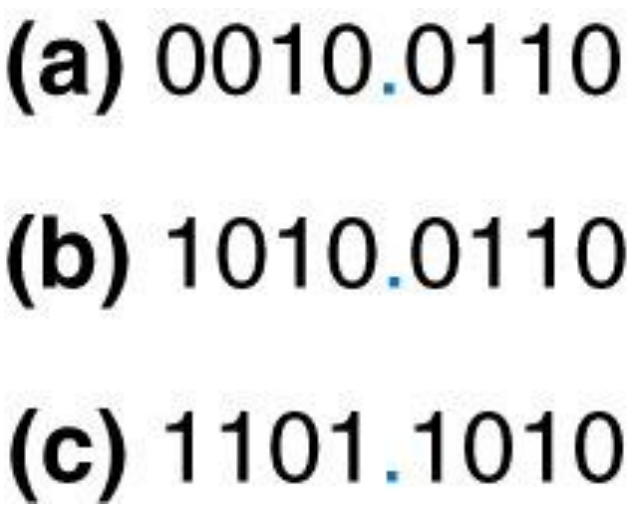


Figure 5.22 Fixed-point representation of □2.375: (a) absolute value, (b) sign and magnitude, (c) two's complement

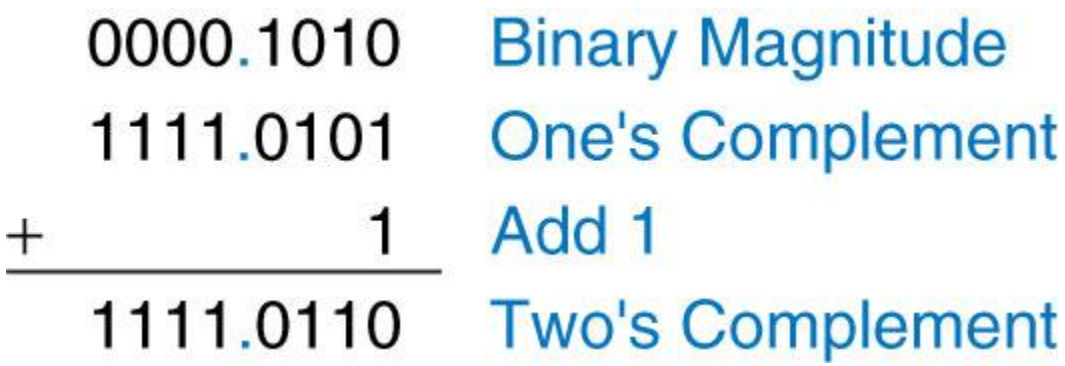


Figure 5.23 Fixed-point two's complement conversion

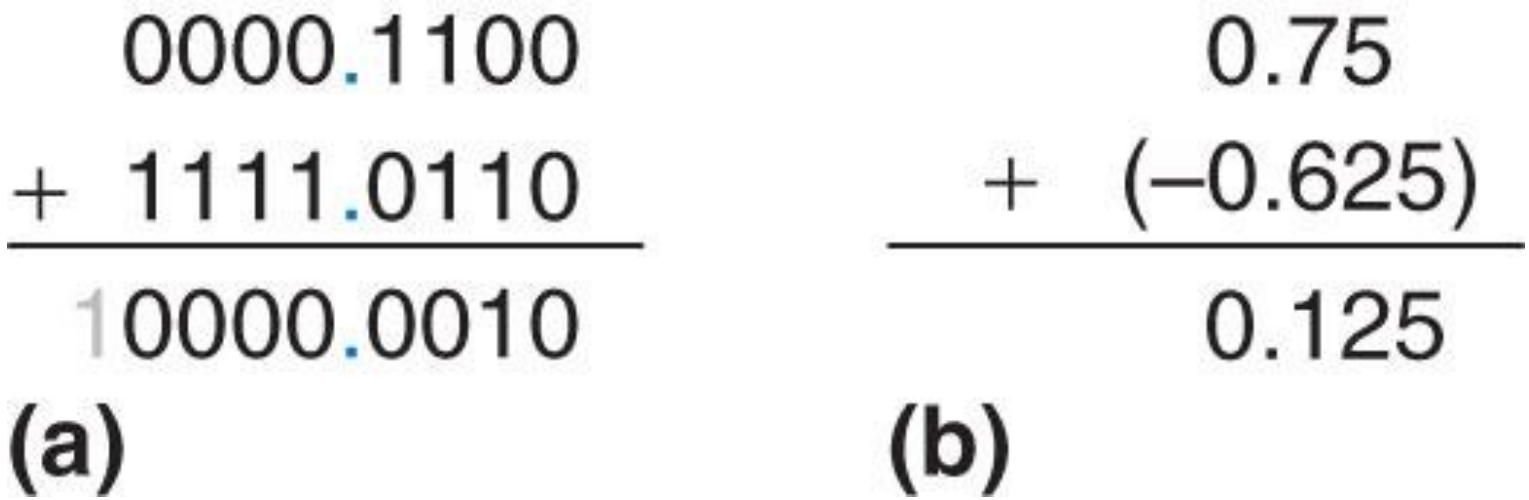


Figure 5.24 Addition: (a) binary fixed-point, (b) decimal equivalent

Figure 5.25 Floating-point numbers

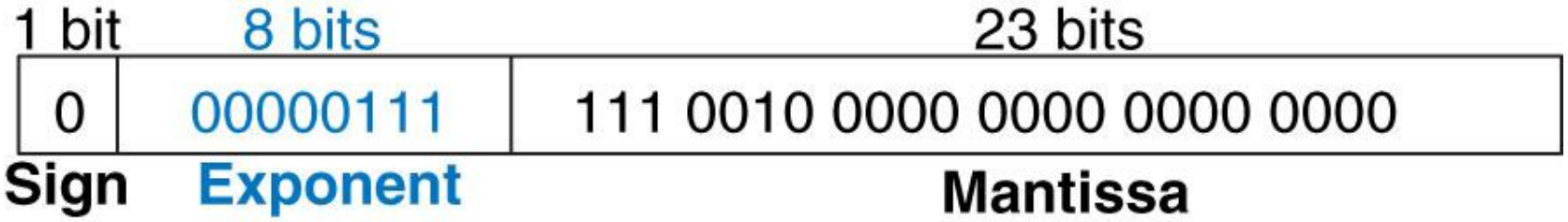


Figure 5.26 32-bit floating-point version 1

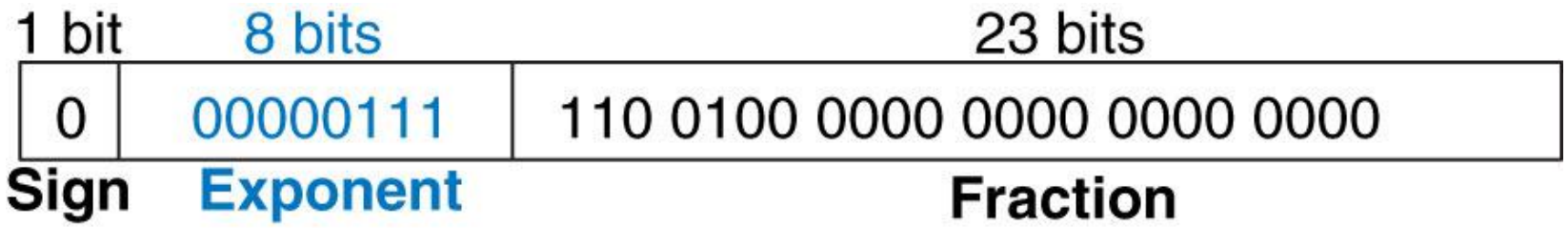


Figure 5.27 Floating-point version 2

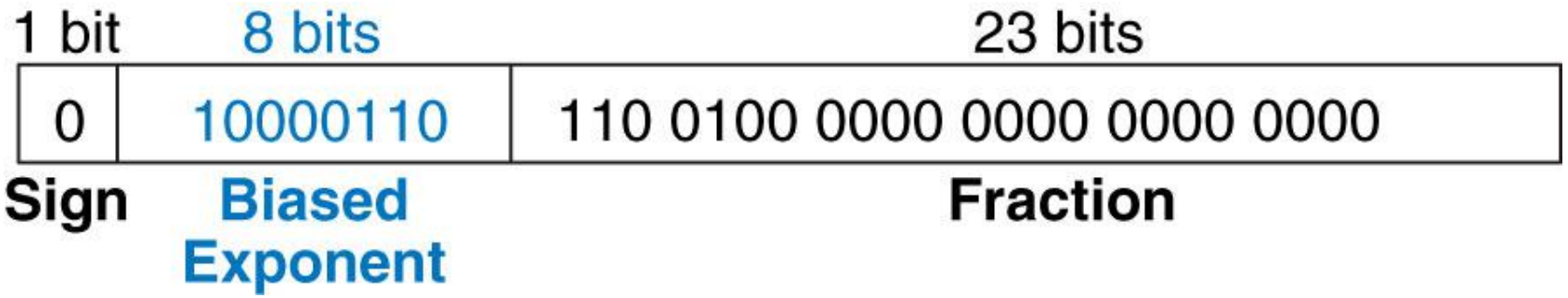


Figure 5.28 IEEE 754 floating-point notation

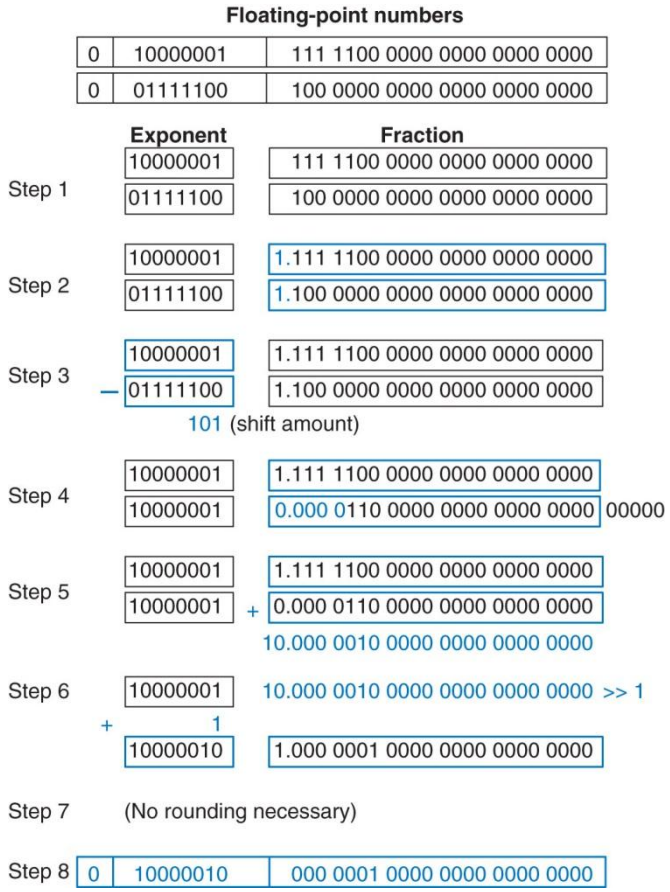


Figure 5.29 Floating-point addition

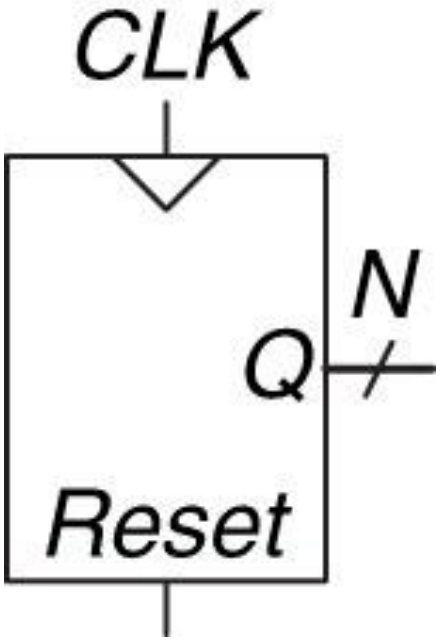


Figure 5.30 Counter symbol

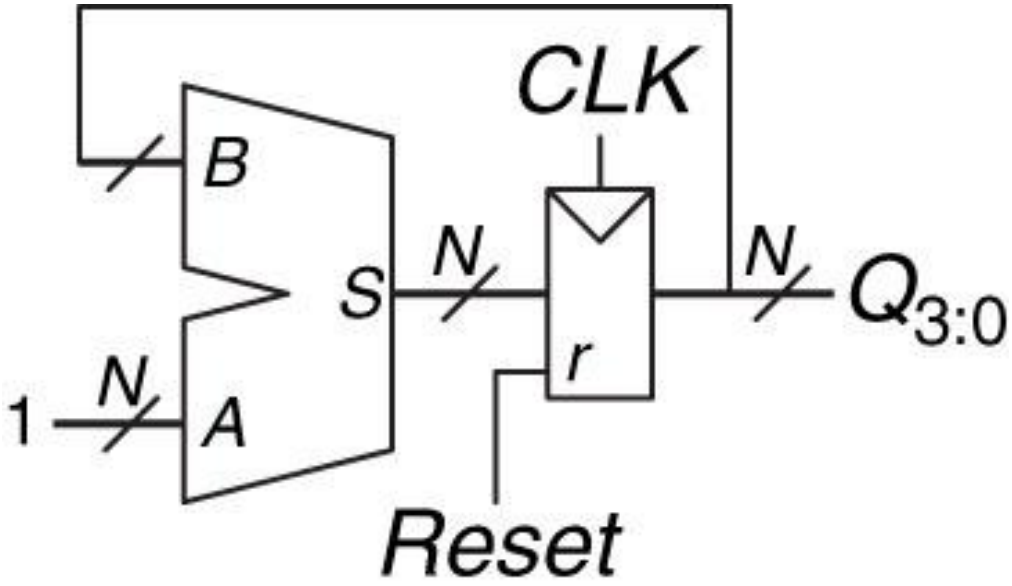


Figure 5.31 *N*-bit counter

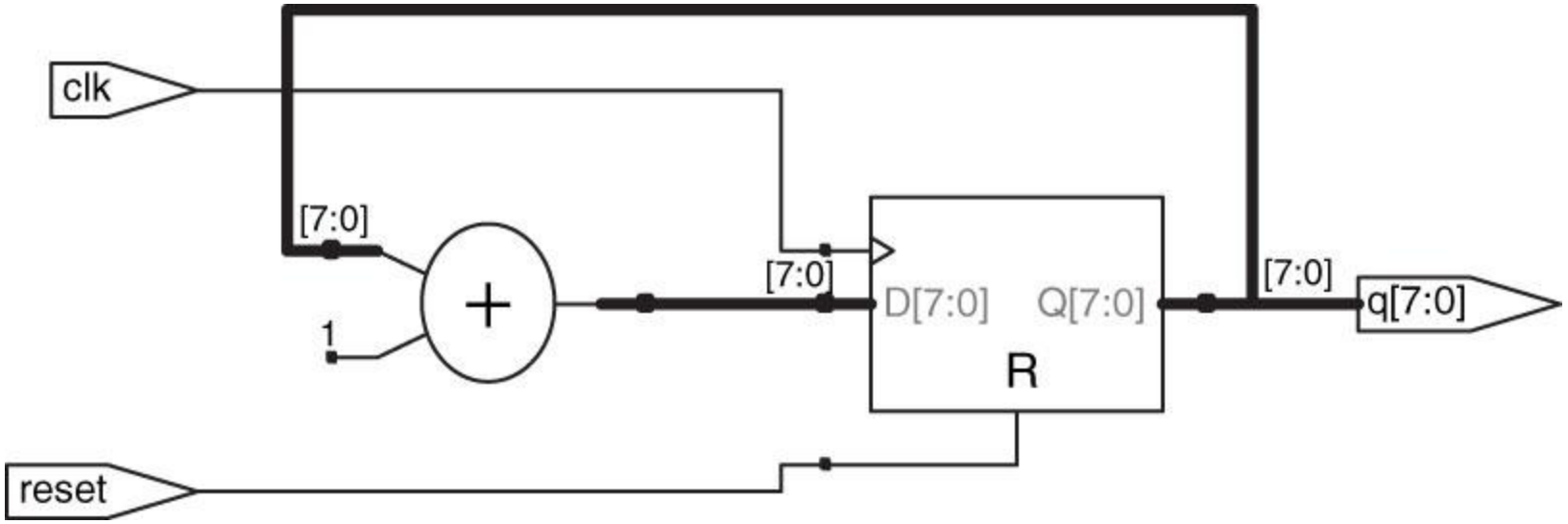


Figure 5.32 Synthesized counter

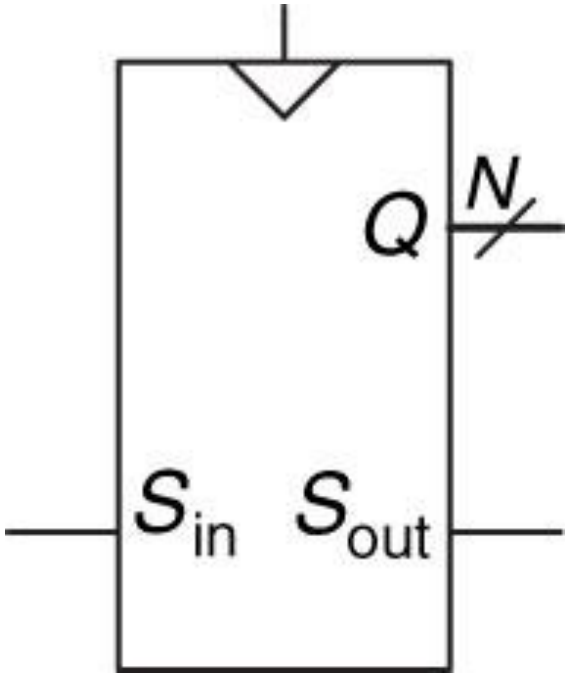


Figure 5.33 Shift register symbol

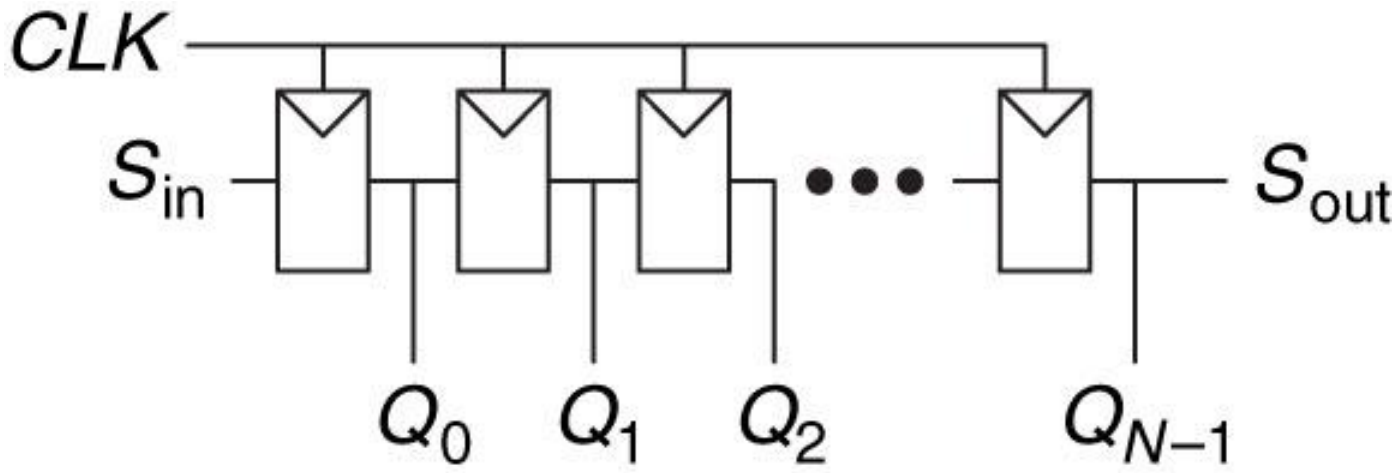


Figure 5.34 Shift register schematic

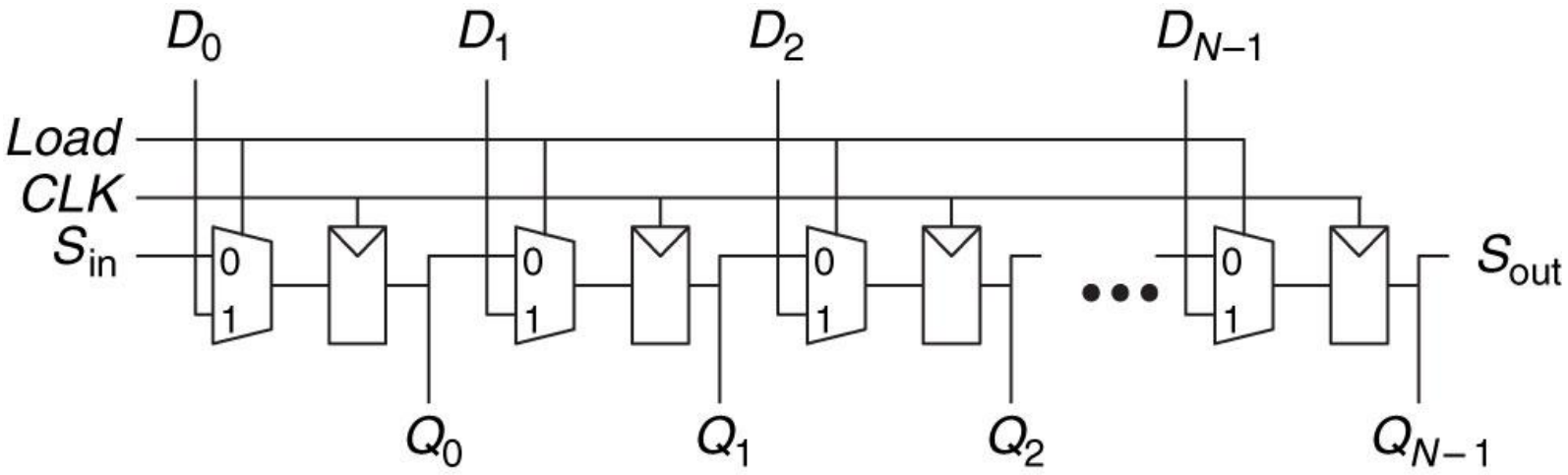


Figure 5.35 Shift register with parallel load

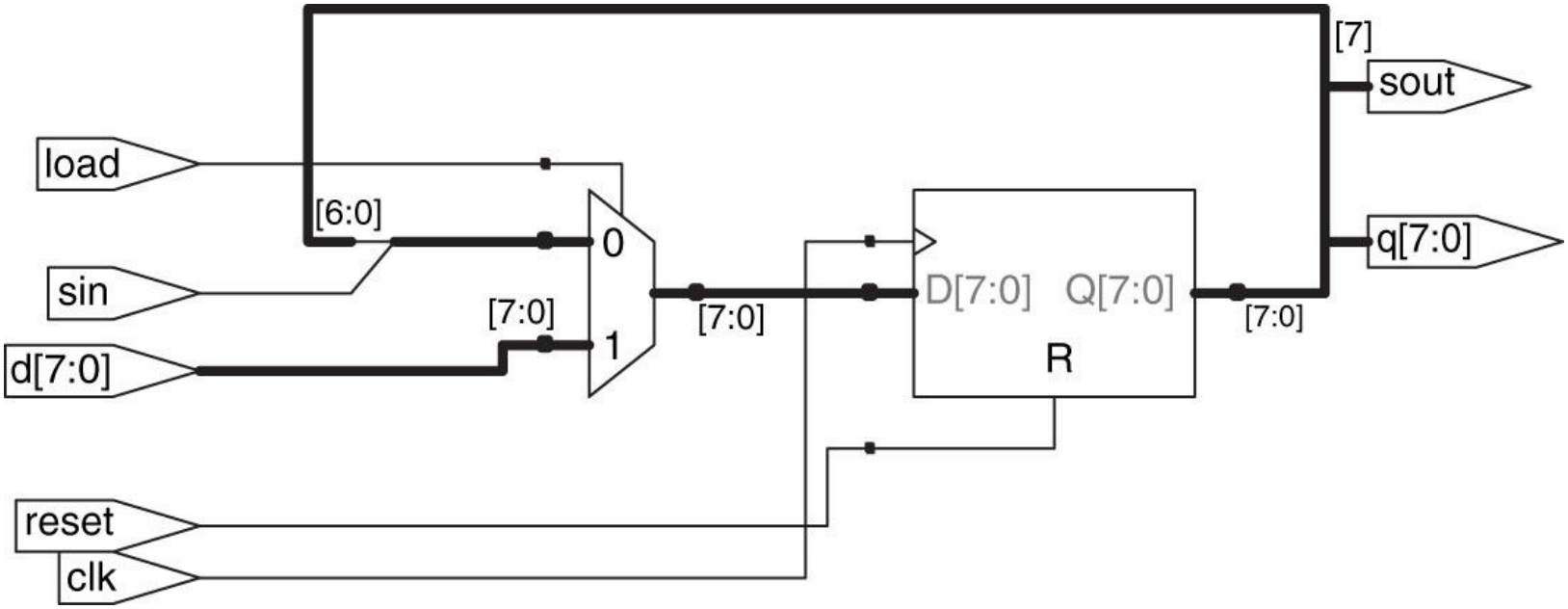


Figure 5.36 Synthesized shiftreg

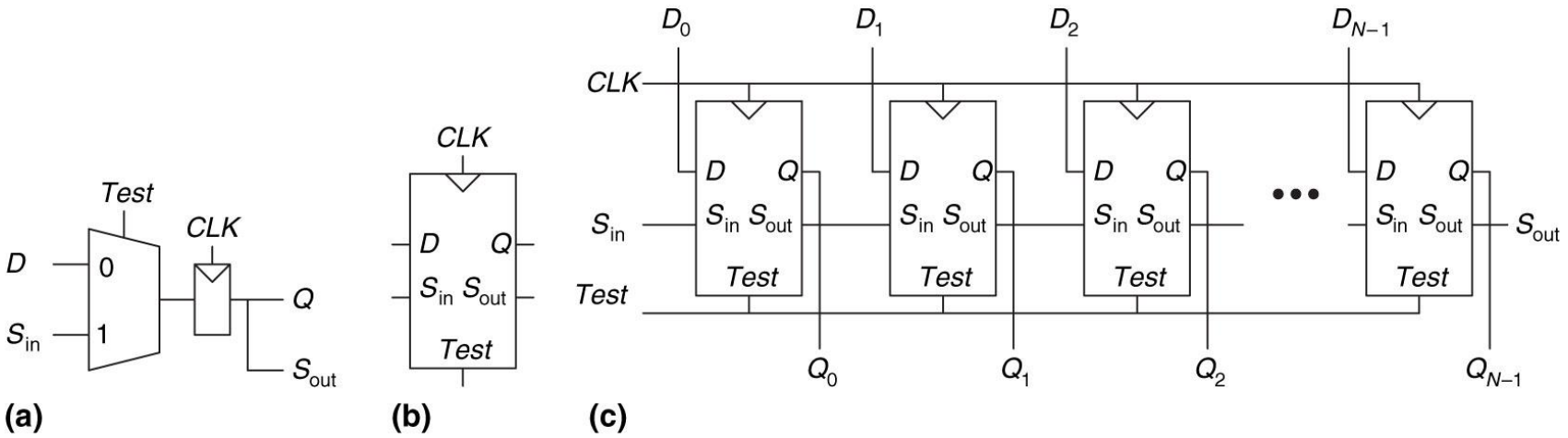


Figure 5.37 Scannable flip-flop: (a) schematic, (b) symbol, and (c) *N*-bit scannable register

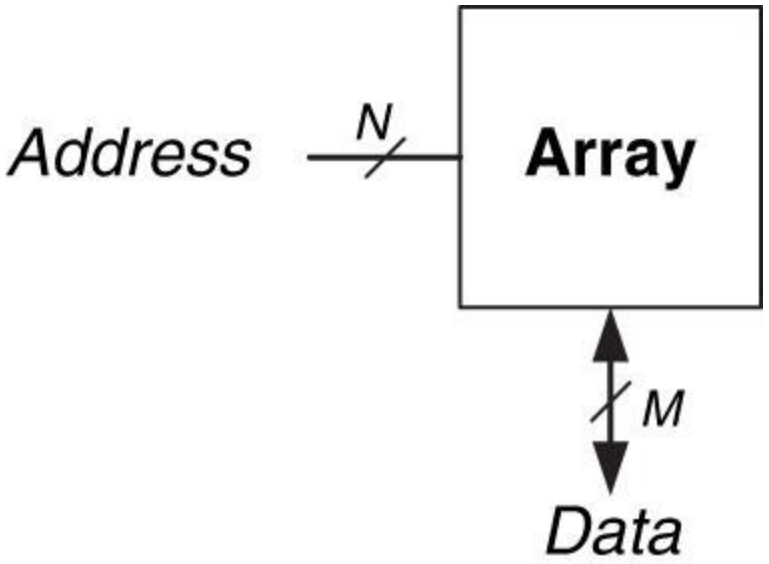


Figure 5.38 Generic memory array symbol

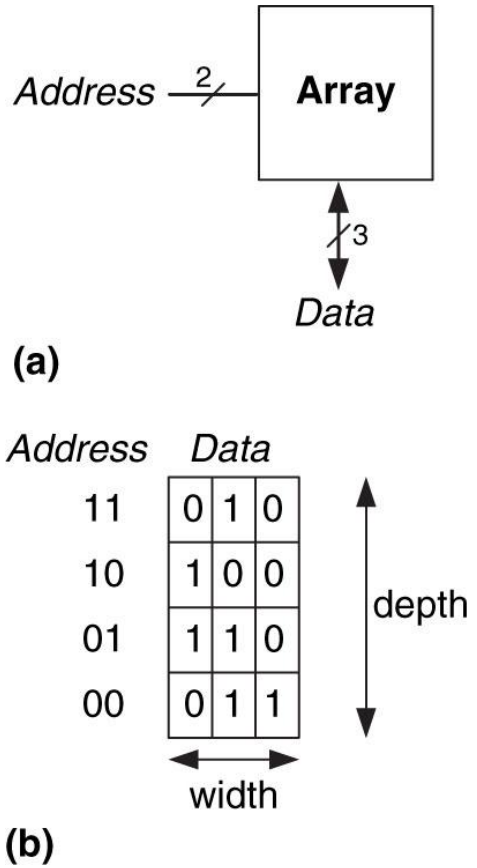


Figure 5.39 4 · 3 memory array: (a) symbol, (b) function

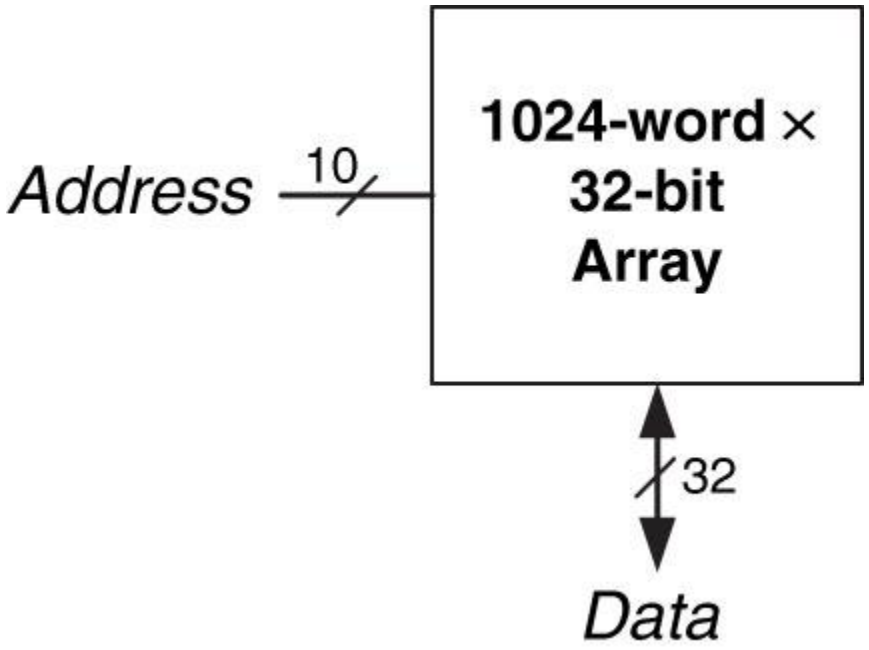


Figure 5.40 32 Kb array: depth = 2^{10} = 1024 words, width = 32 bits

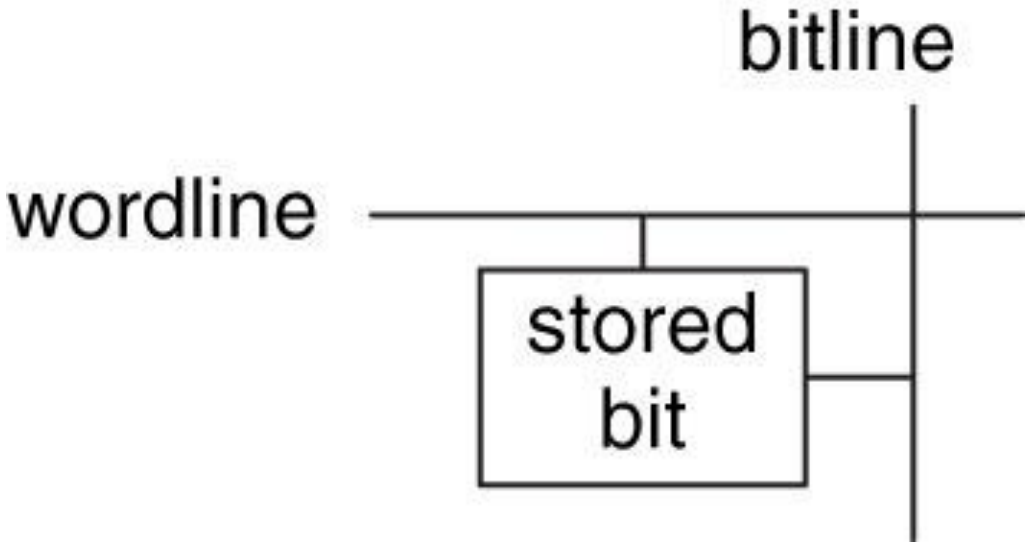


Figure 5.41 Bit cell

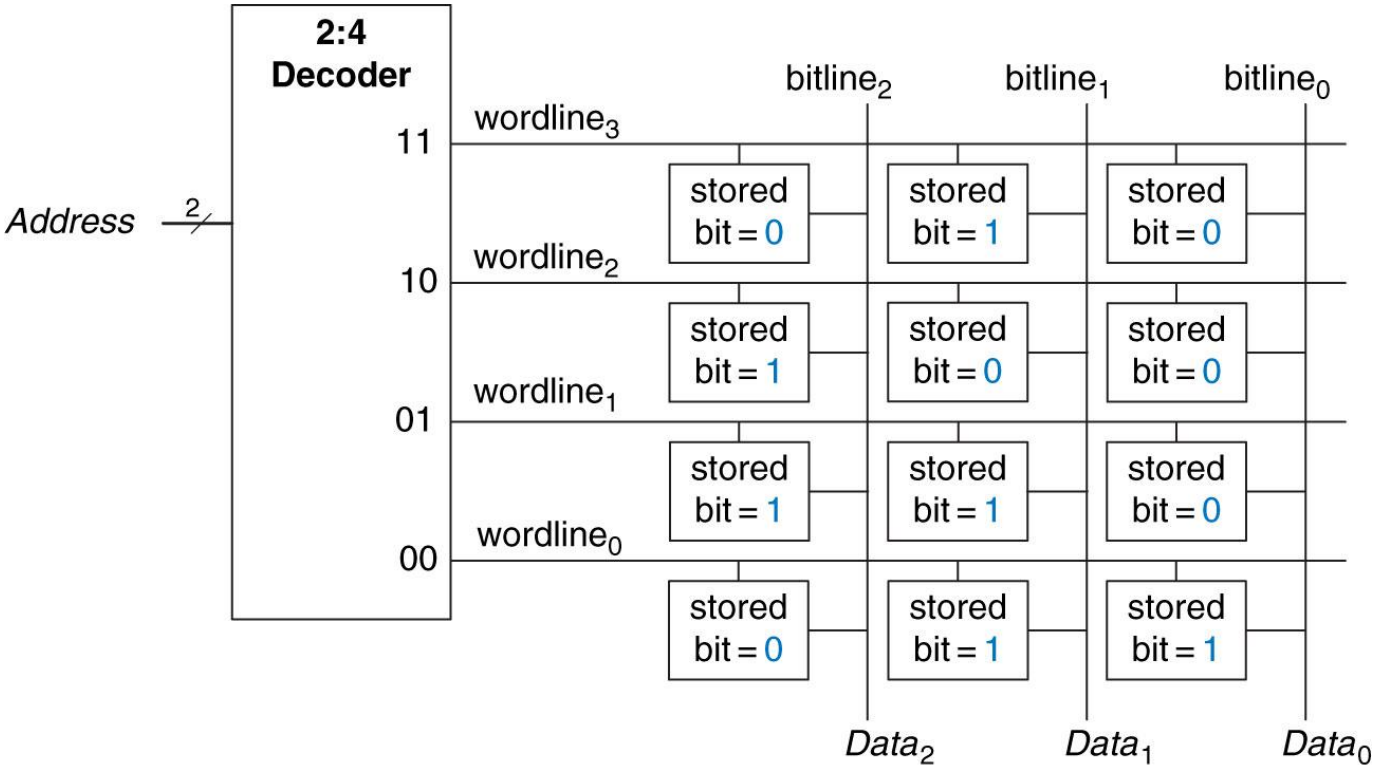


Figure 5.42 4 · 3 memory array

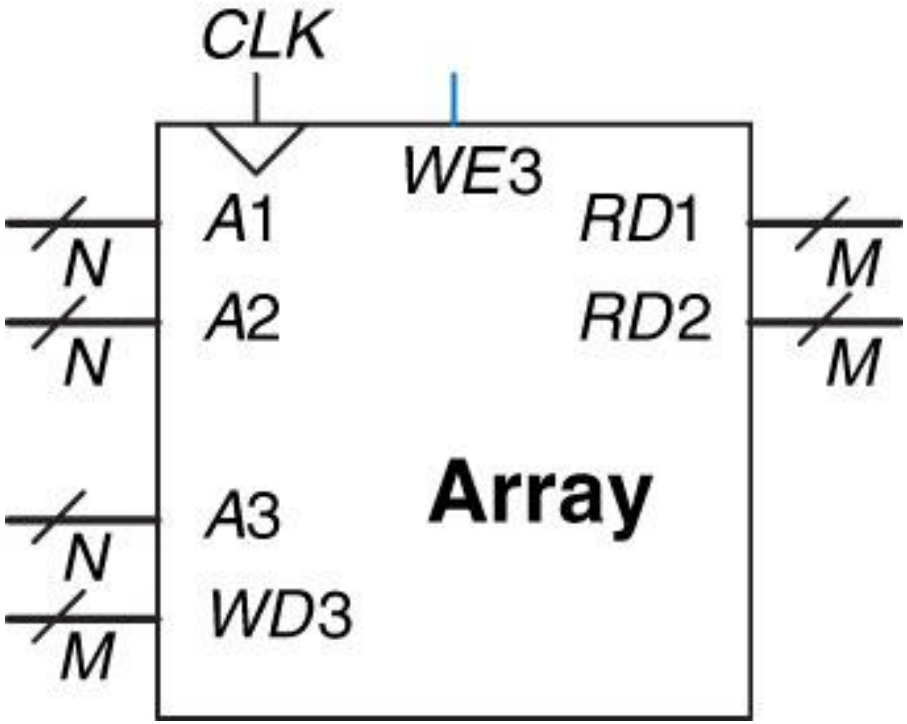


Figure 5.43 Three-ported memory

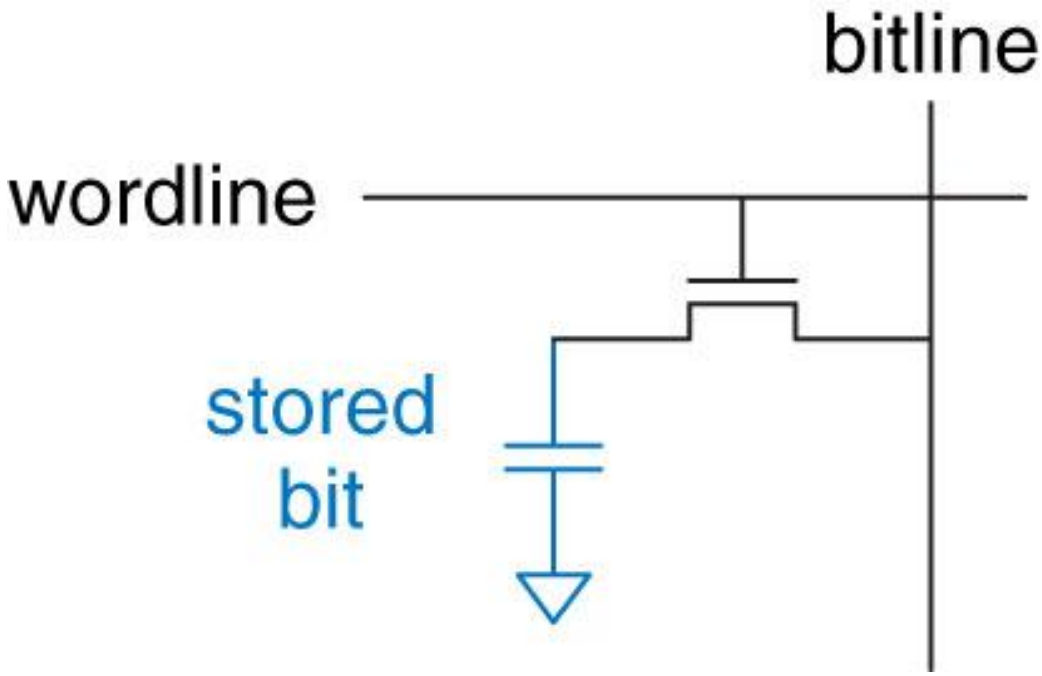


Figure 5.44 DRAM bit cell

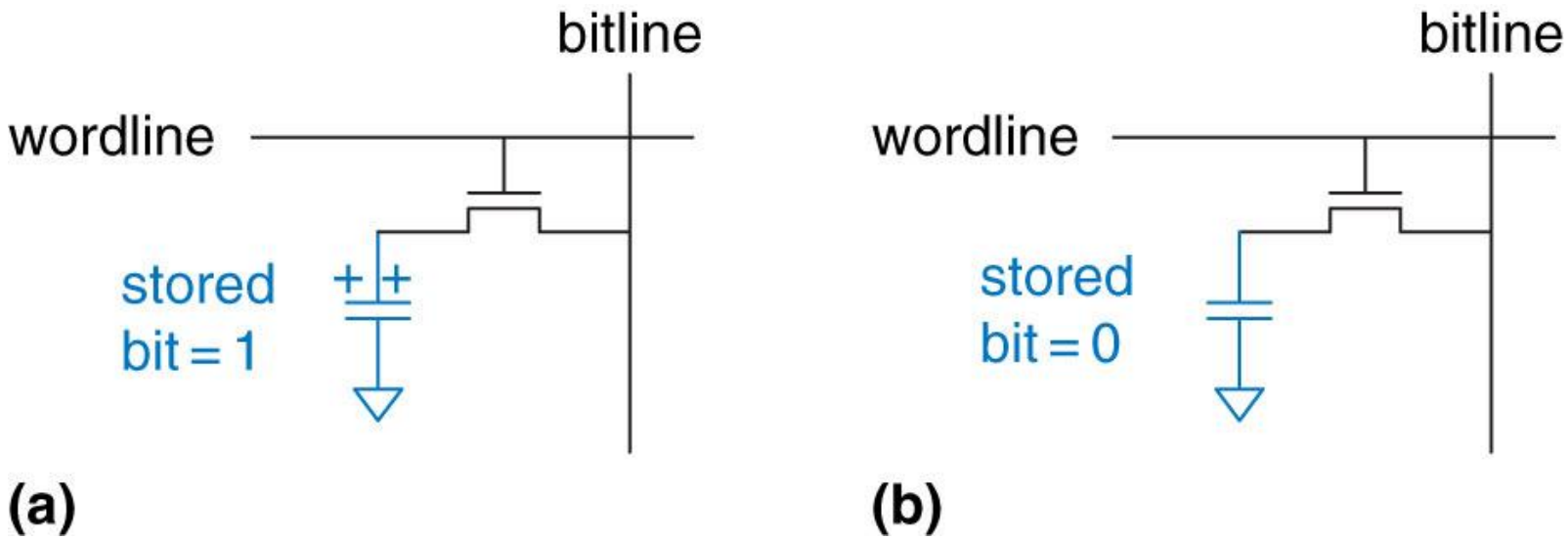


Figure 5.45 DRAM stored values

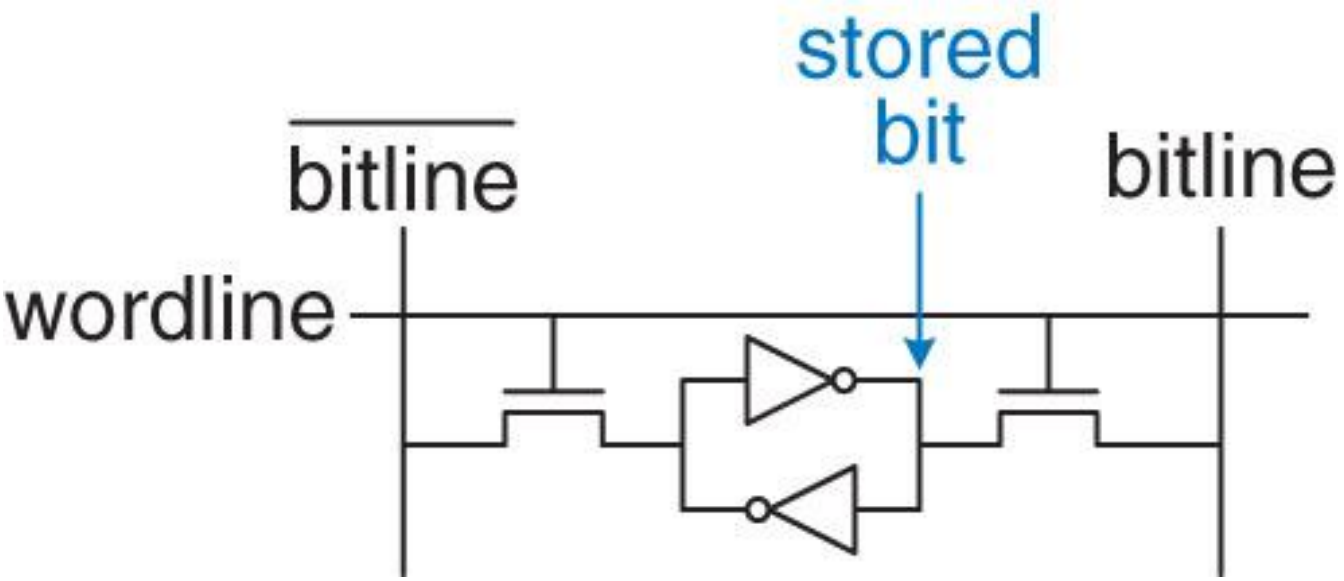


Figure 5.46 SRAM bit cell

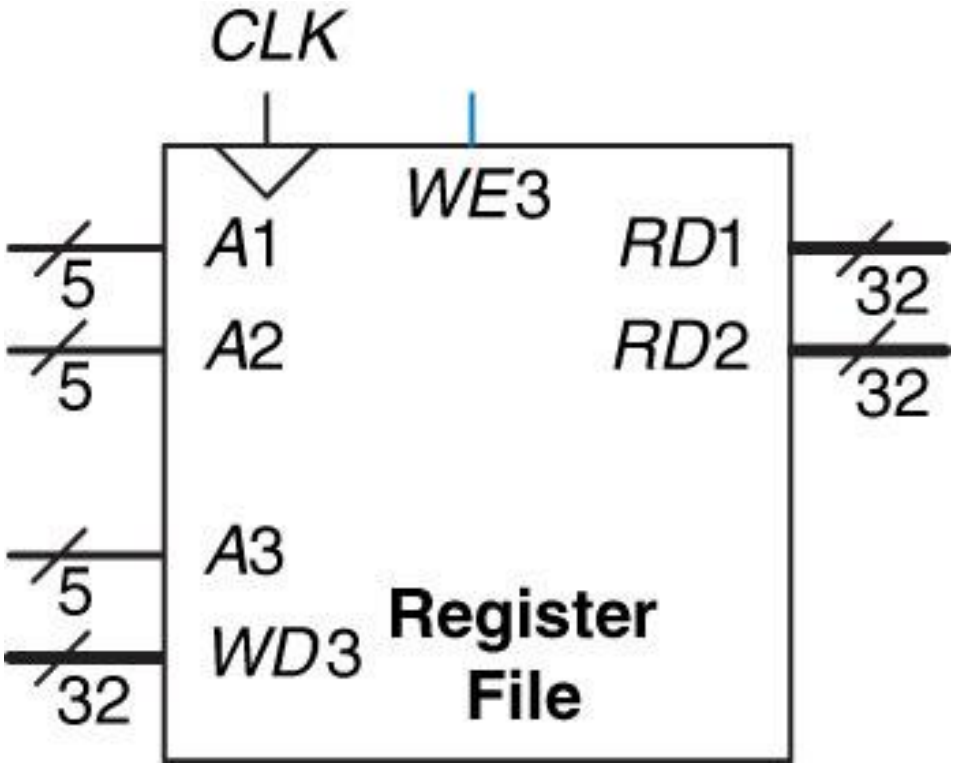


Figure 5.47 32 · 32 register file with two read ports and one write port

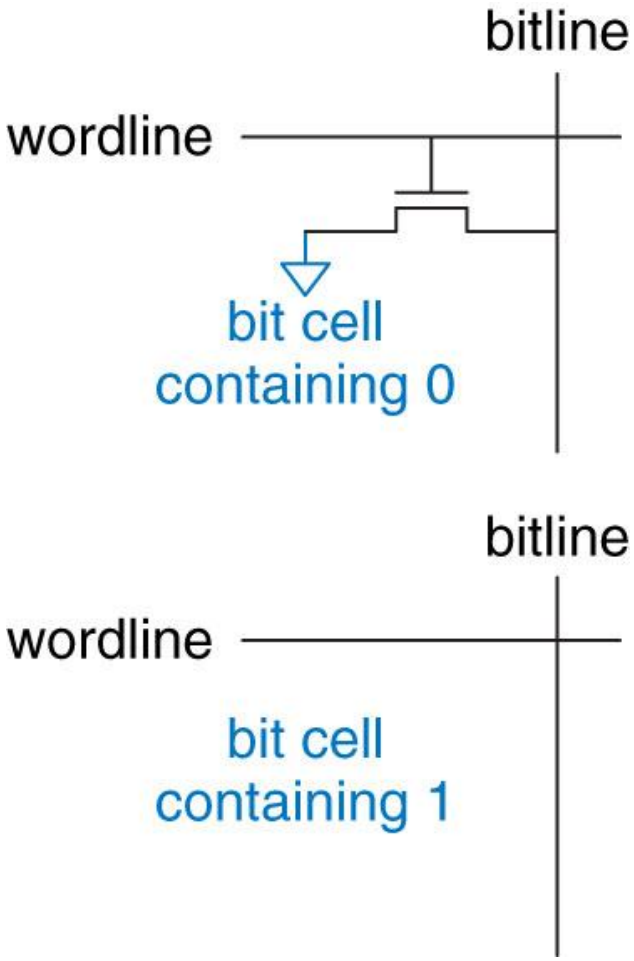


Figure 5.48 ROM bit cells containing 0 and 1

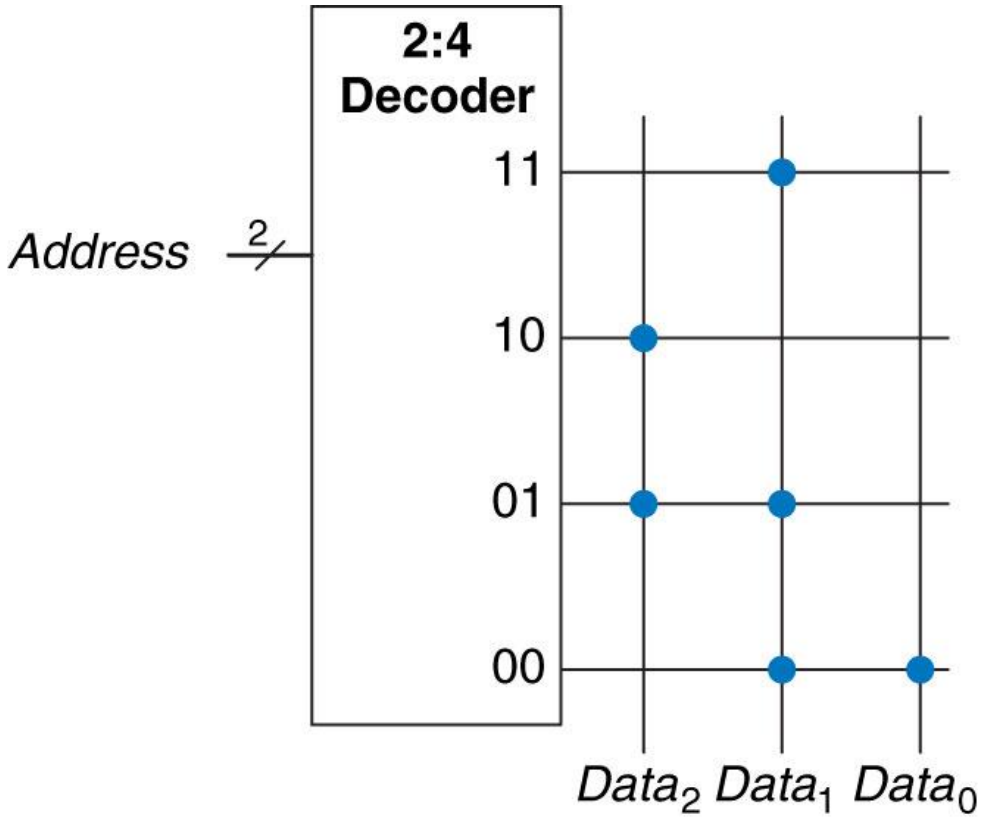


Figure 5.49 4 · 3 ROM: dot notation

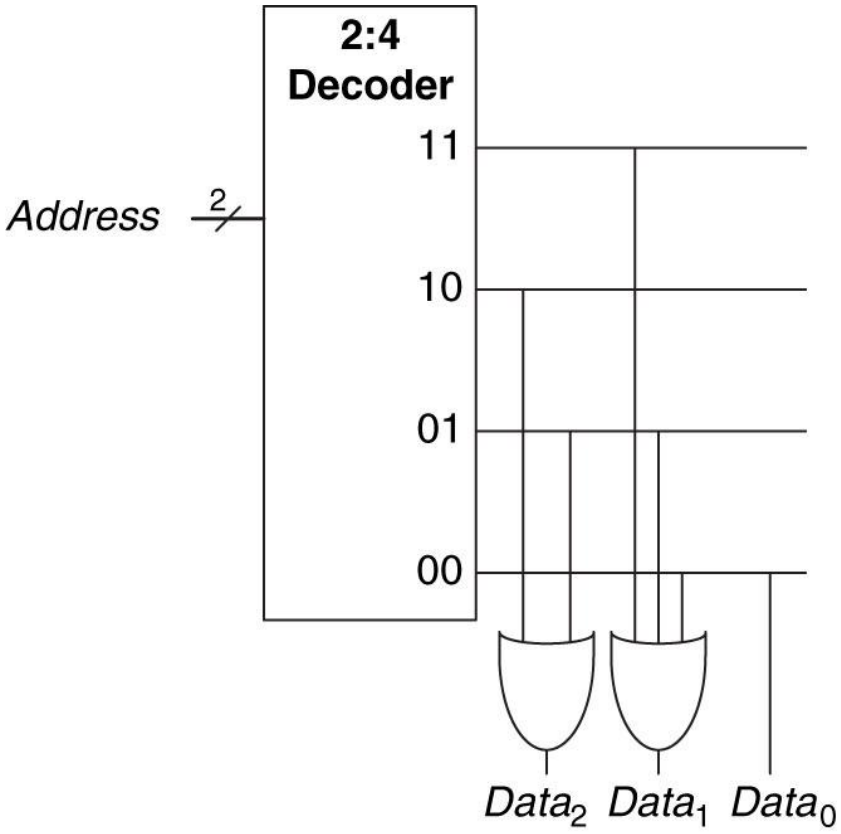


Figure 5.50 4 · 3 ROM implementation using gates

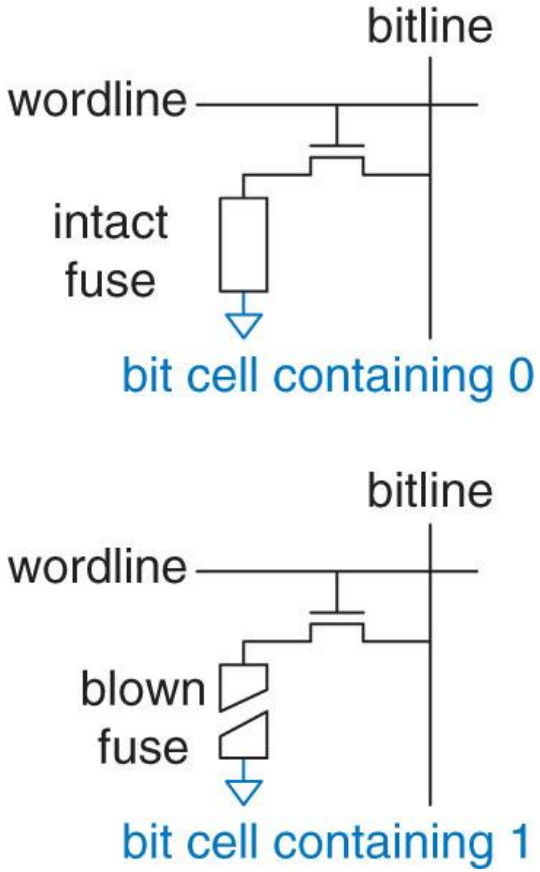


Figure 5.51 Fuse-programmable ROM bit cell

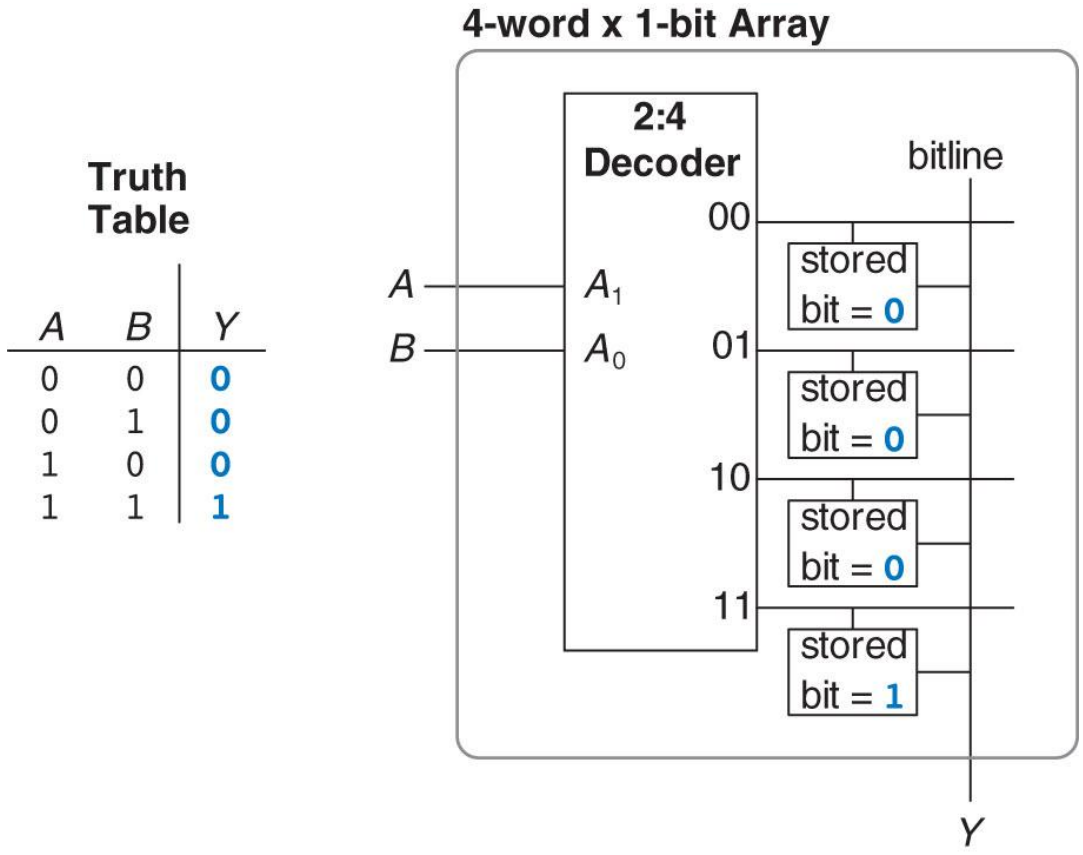


Figure 5.52 4-word · 1-bit memory array used as a lookup table

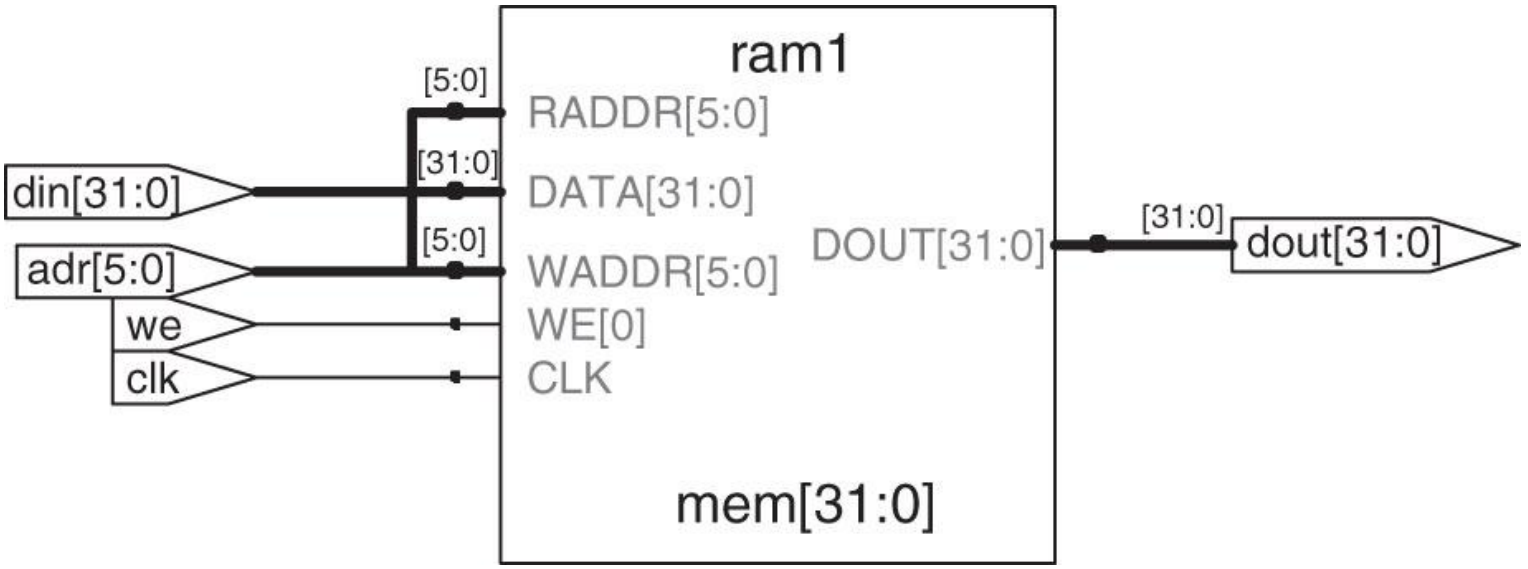


Figure 5.53 Synthesized ram

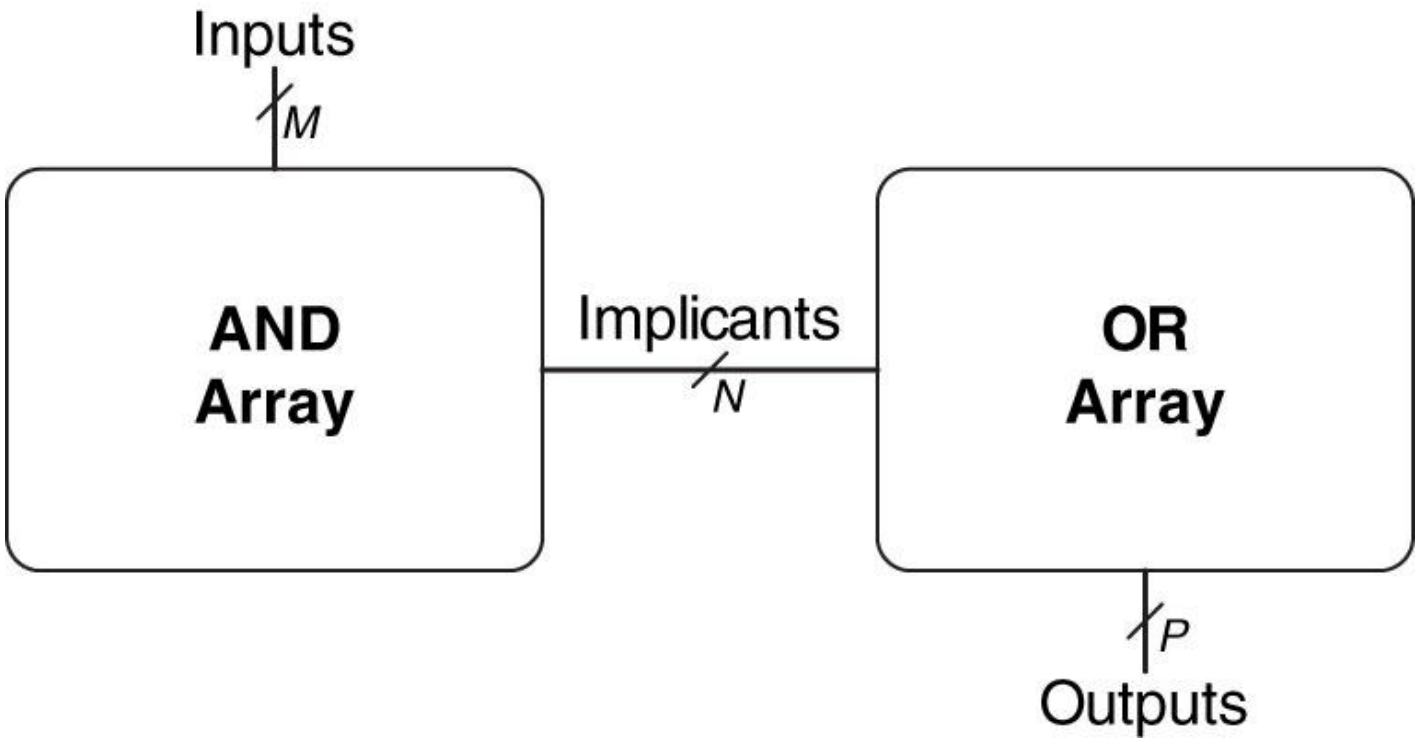


Figure 5.54 *M* · *N* · *P*-bit PLA

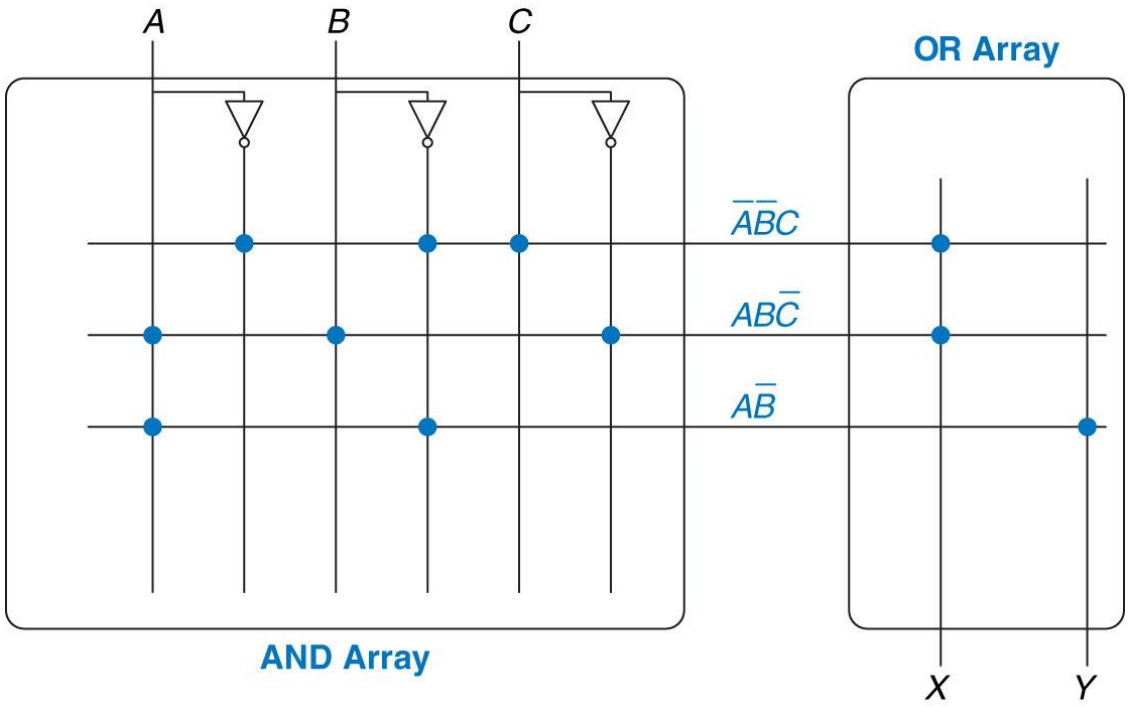


Figure 5.55 3 · 3 · 2-bit PLA: dot notation

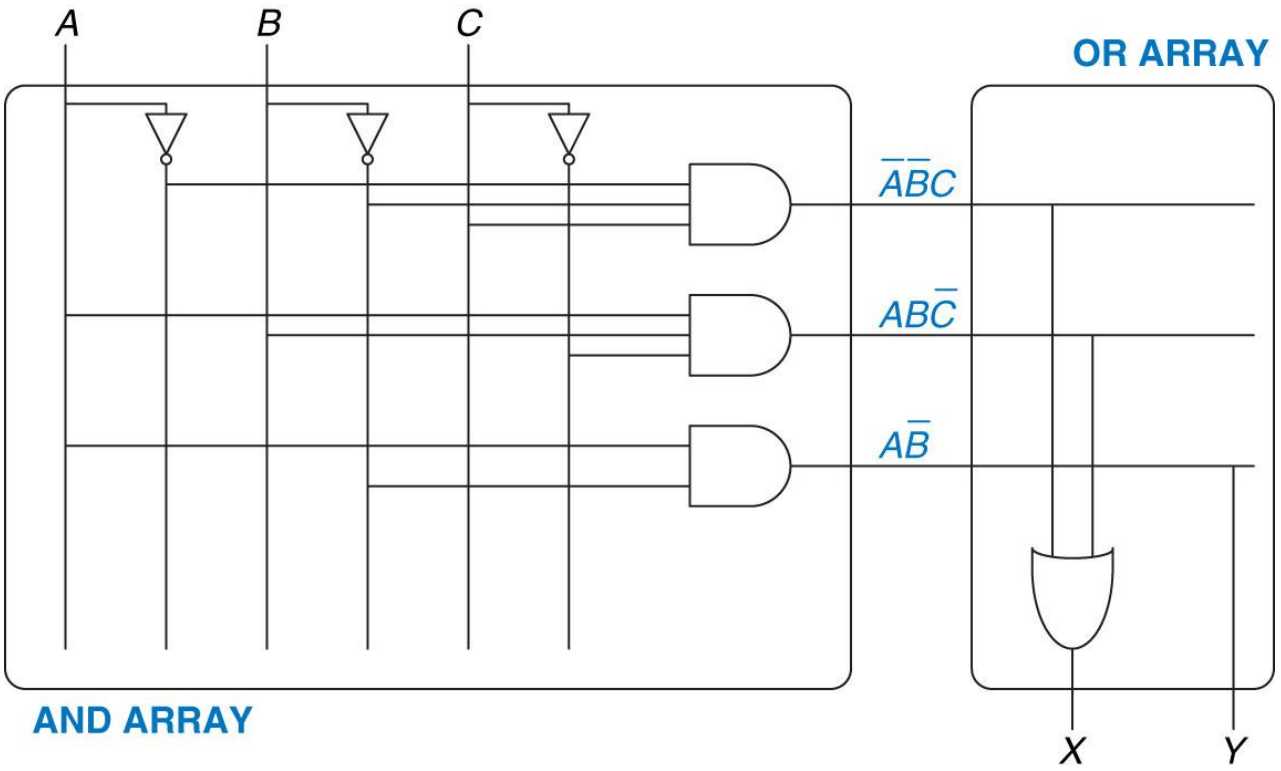


Figure 5.56 3 · 3 · 2-bit PLA using two-level logic

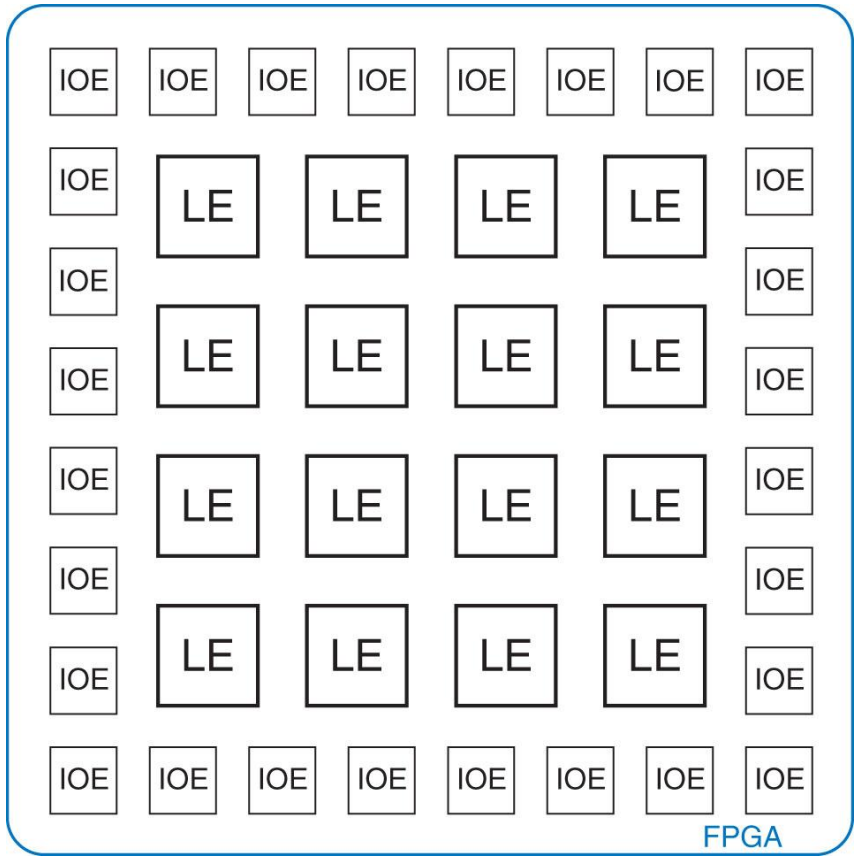


Figure 5.57 General FPGA layout

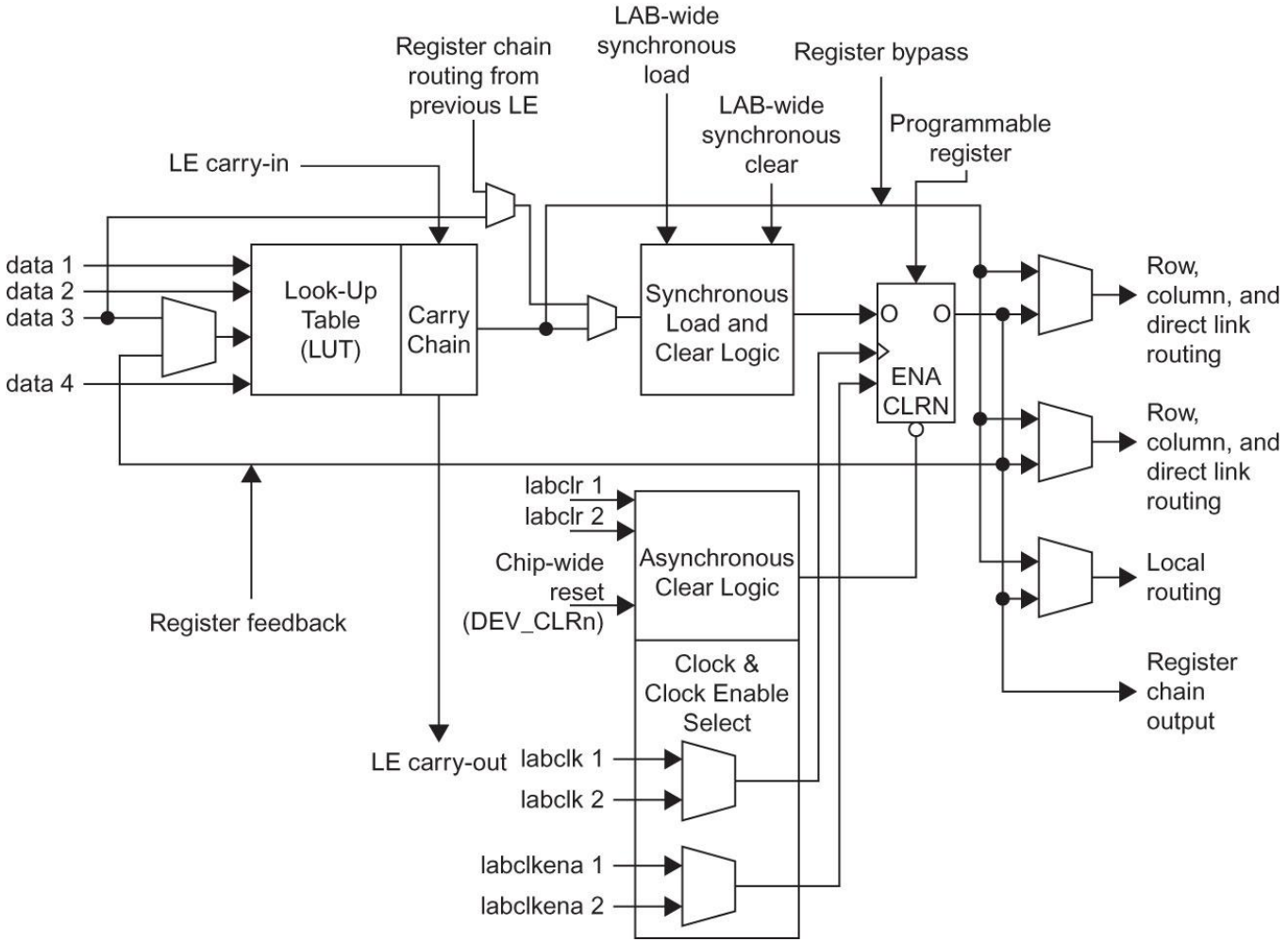


Figure 5.58 Cyclone IV Logic Element (LE)
(Reproduced with permission from the Altera Cyclone· IV Handbook · 2010 Altera Corporation.)

(A) data 1	(B) data 2	(C) data 3	data 4	(X) LUT output
0	0	0	X	0
0	0	1	X	1
0	1	0	X	0
0	1	1	X	0
1	0	0	X	0
1	0	1	X	0
1	1	0	X	1
1	1	1	X	0

(A) data 1	(B) data 2	data 3	data 4	(Y) LUT output
0	0	X	X	0
0	1	X	X	0
1	0	X	X	1
1	1	X	X	0

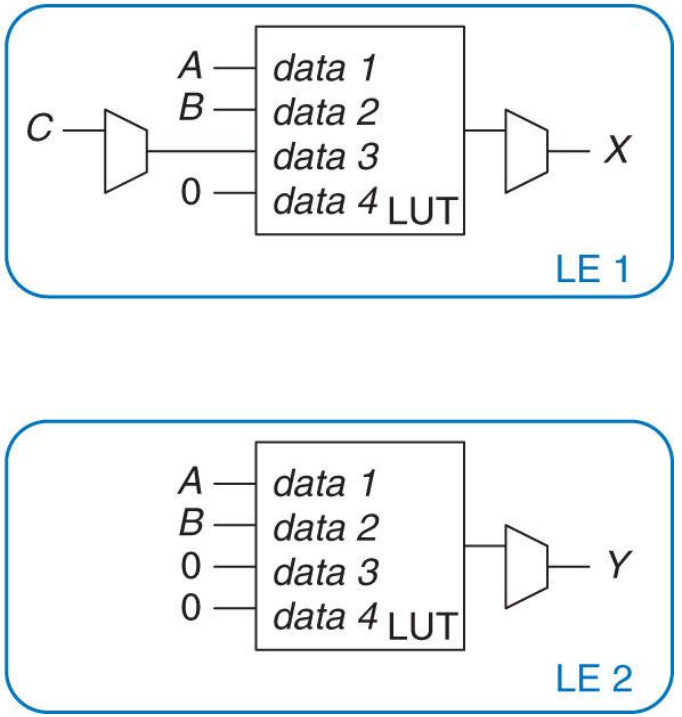


Figure 5.59 LE configuration for two functions of up to four inputs each

(J) data 1	(K) data 2	(L) data 3	(M) data 4	(X) LUT output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(P) data 1	(Q) data 2	(R) data 3	(X) data 4	(Y) LUT output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

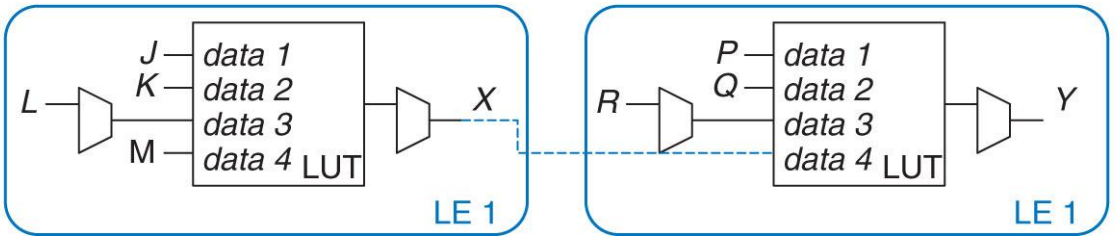


Figure 5.60 LE configuration for one function of more than four inputs

data 1	data 2	(S_0) data 3	(S_1) data 4	(S_0') LUT output
X	X	0	0	1
X	X	0	1	0
X	X	1	0	0
X	X	1	1	0

data 1	data 2	(S_1) data 3	(S_0) data 4	(S_1') LUT output
X	X	0	0	0
X	X	0	1	1
X	X	1	0	0
X	X	1	1	0

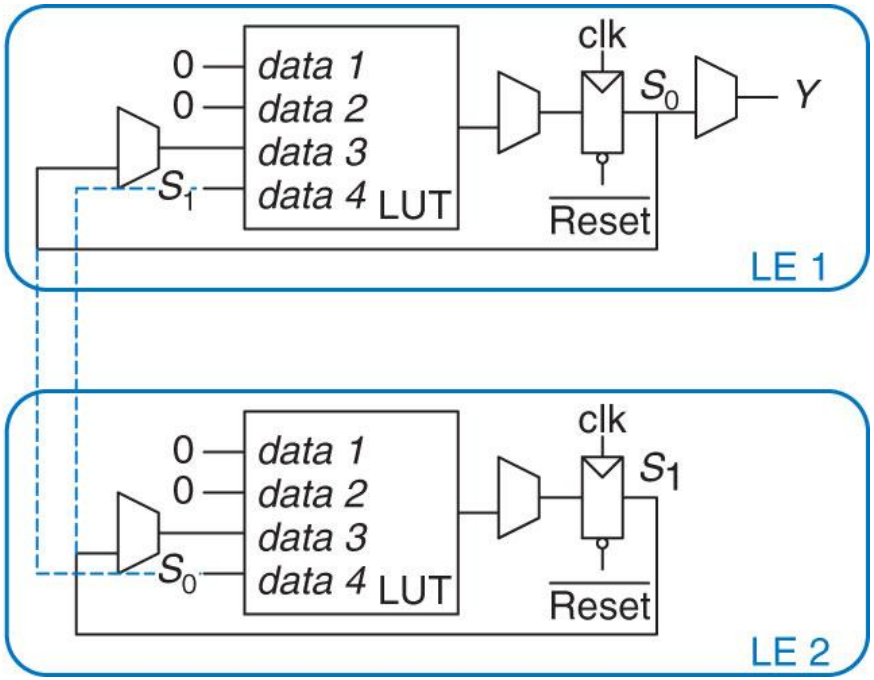


Figure 5.61 LE configuration for FSM with two bits of state

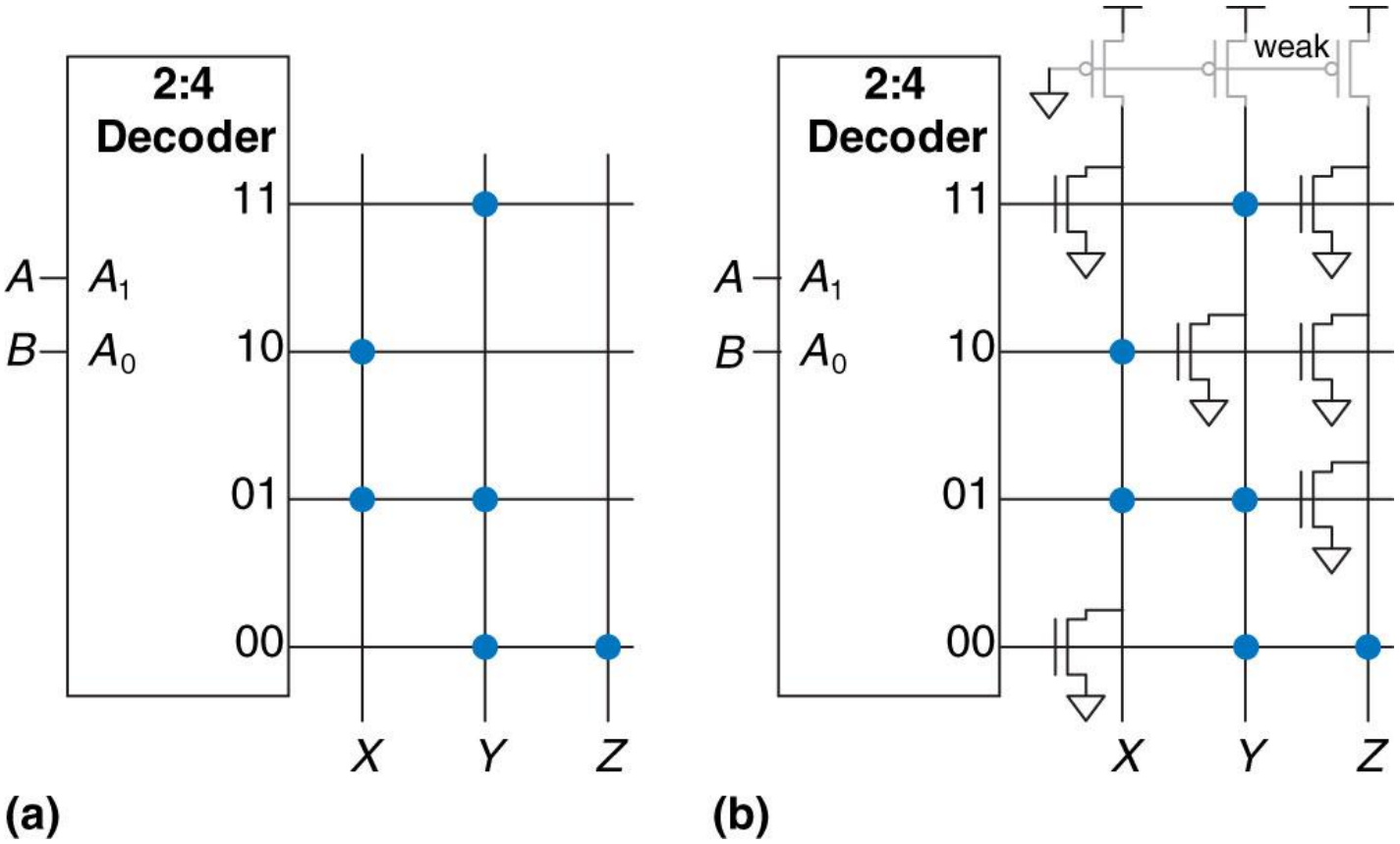


Figure 5.62 ROM implementation: (a) dot notation, (b) pseudo-nMOS circuit

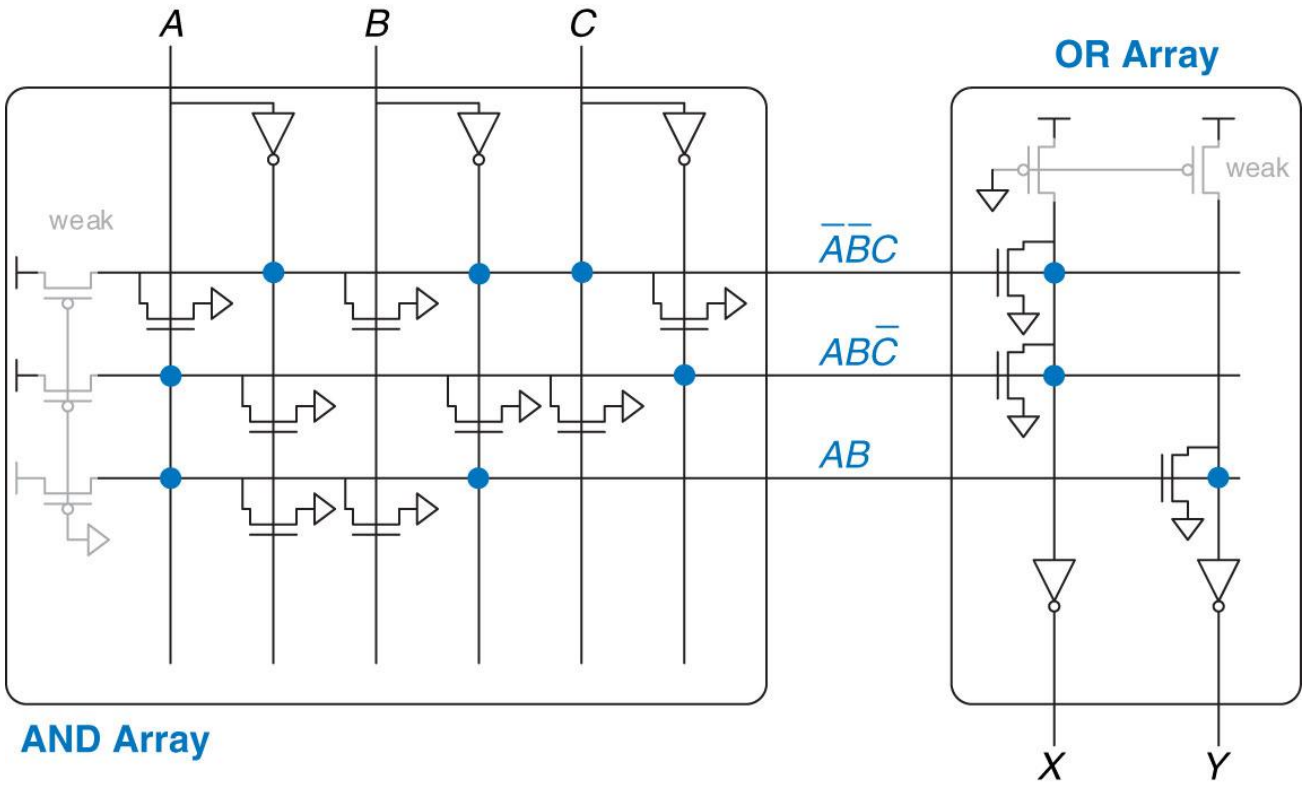


Figure 5.63 3 · 3 · 2-bit PLA using pseudo-nMOS circuits

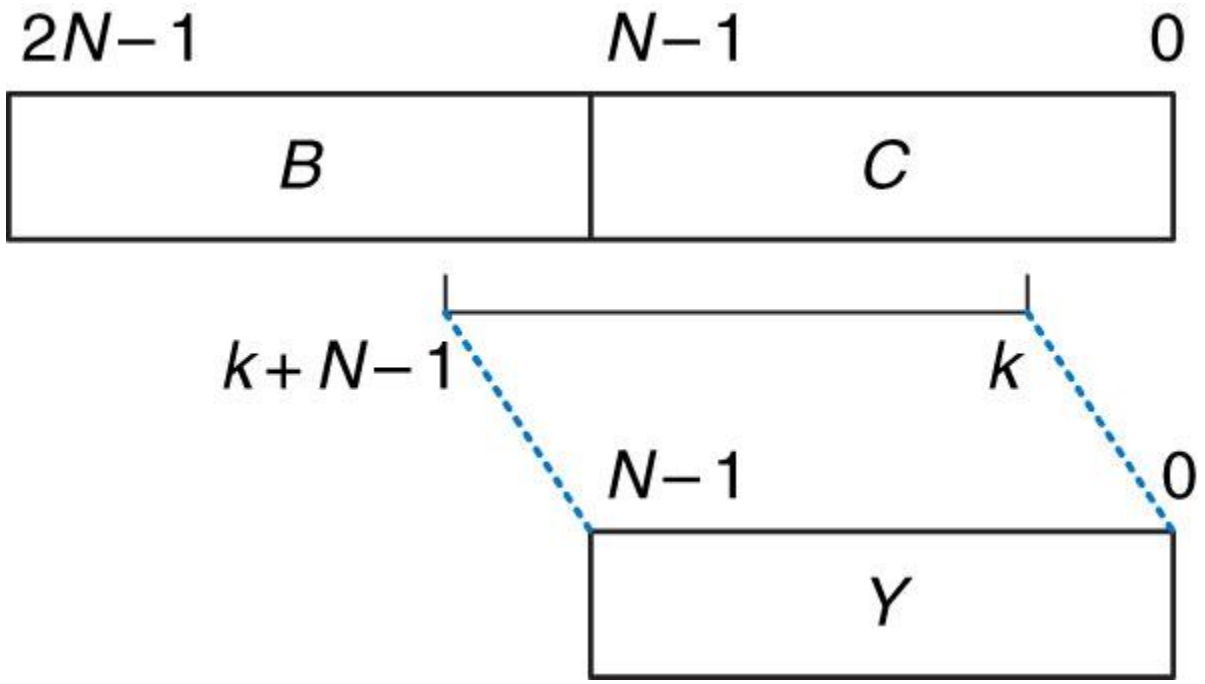


Figure 5.64 Funnel shifter

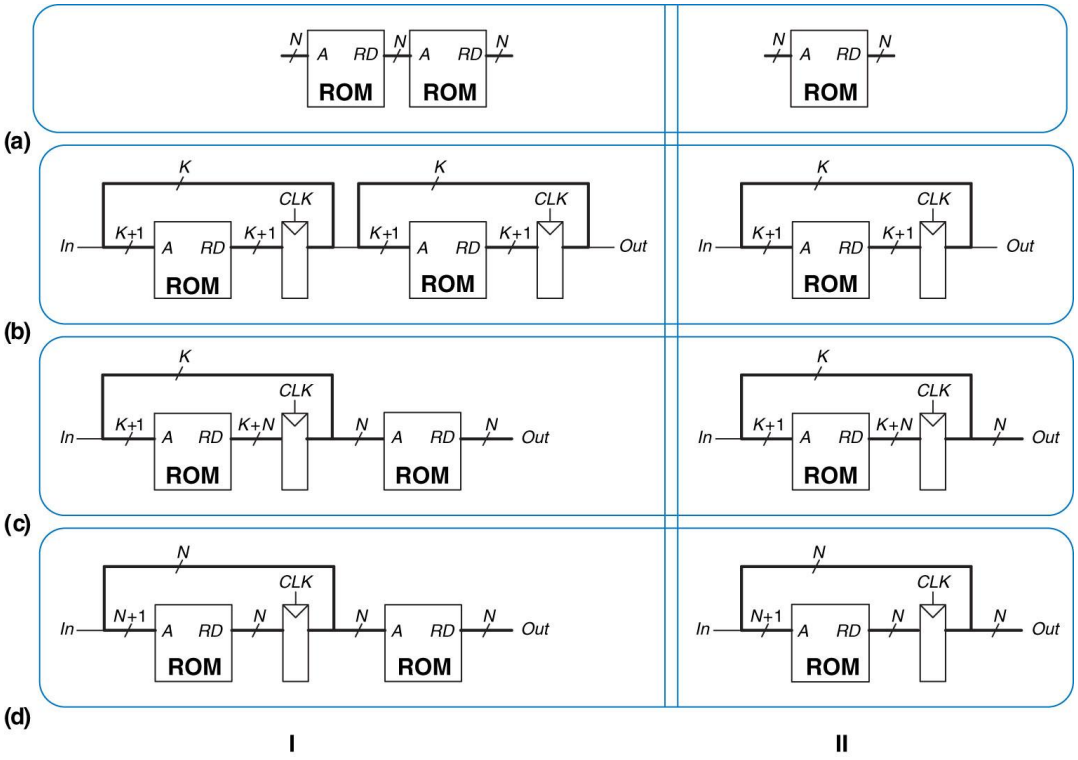


Figure 5.65 ROM circuits

Figure M 01

Figure M 02

Figure M 03

Figure M 04

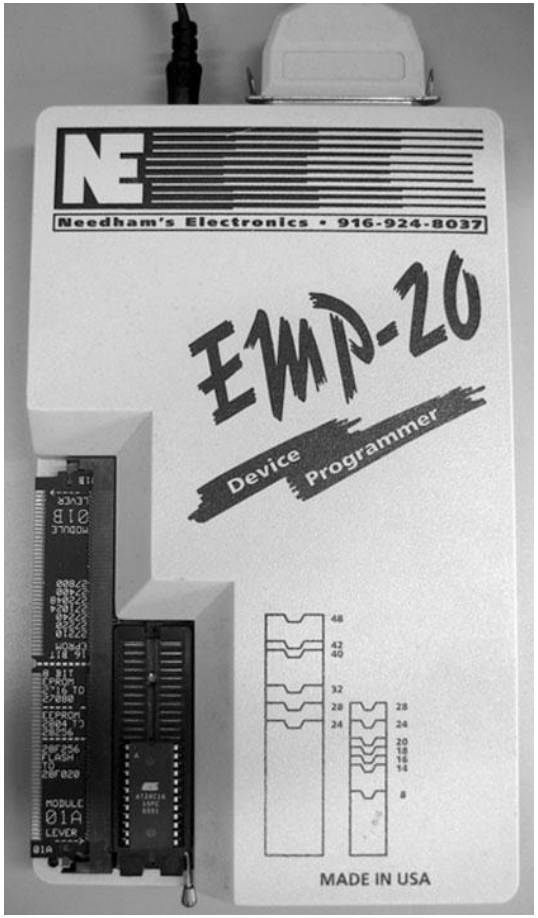
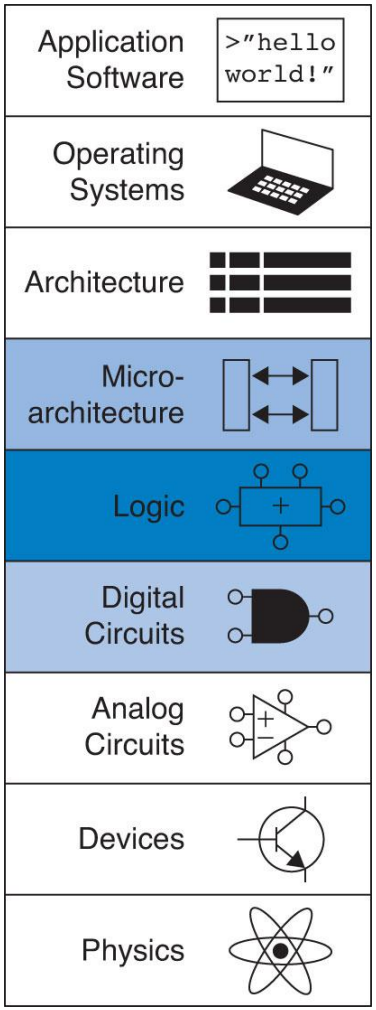


Figure M 05

Figure M 06



UNN Figure 1