Chapter 8

Chip Design

999999-1
VAI 10/28/02

Datapaht and Control

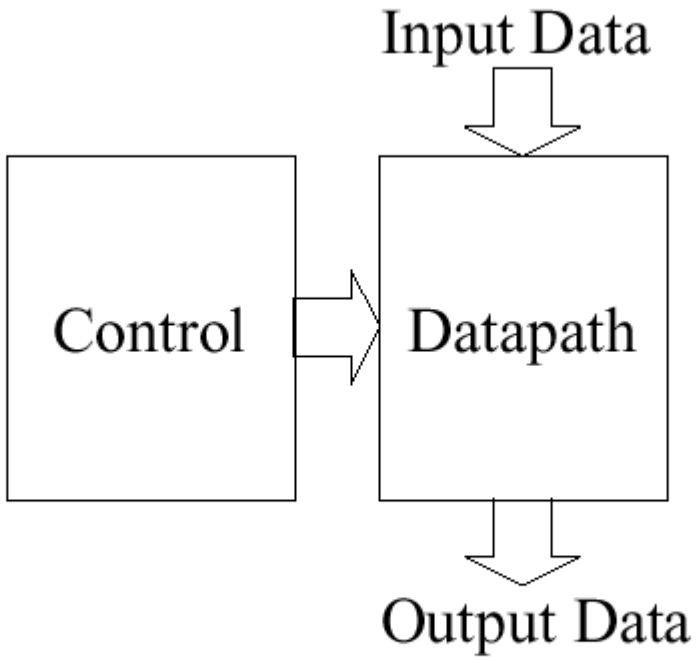

Fig. 8.1 Relationship between datapath and control units.

Finite-State Machine Controlled Datapath

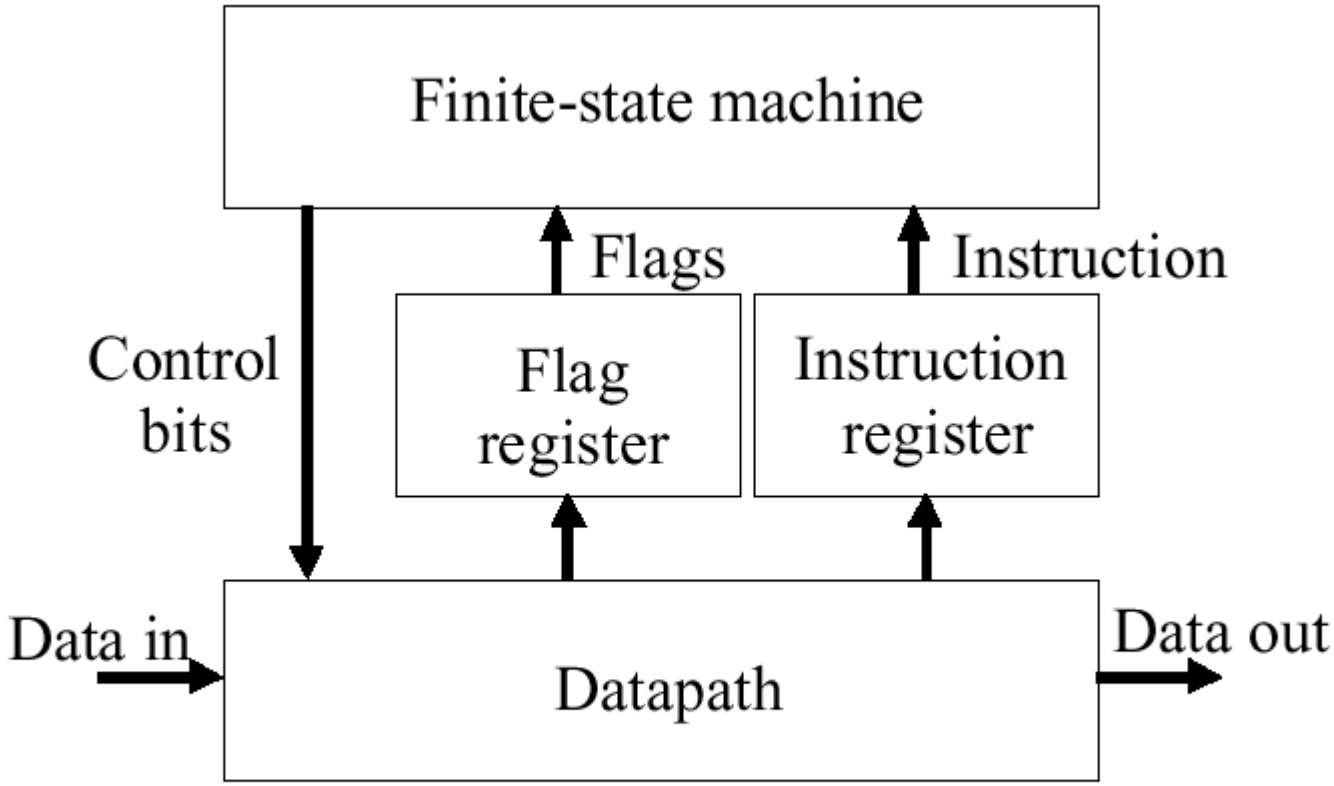

Fig. 8.2 Datapath controlled by a finite-state machine control unit.

Microprogrammed Datapath

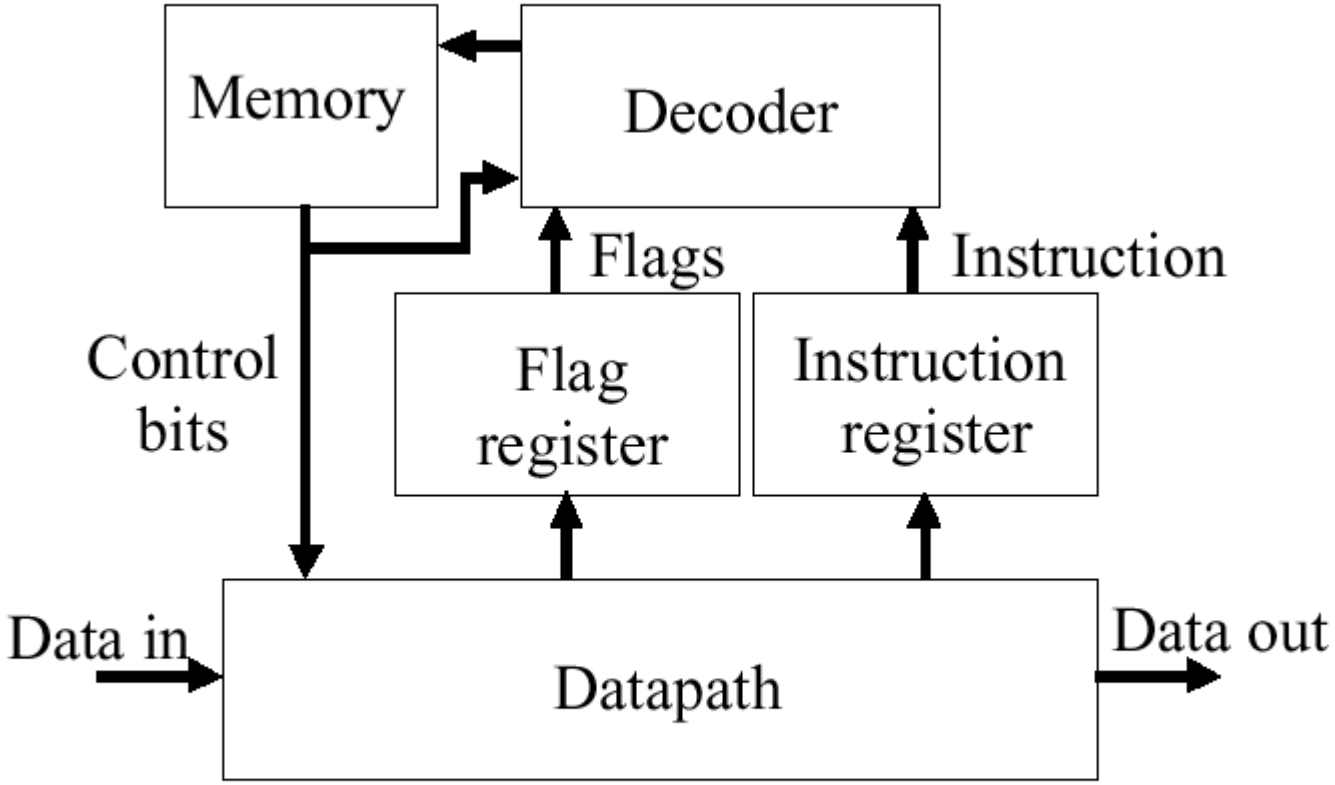

Fig. 8.3 Datapath controlled by a microprogrammed control unit.

SM and Memory

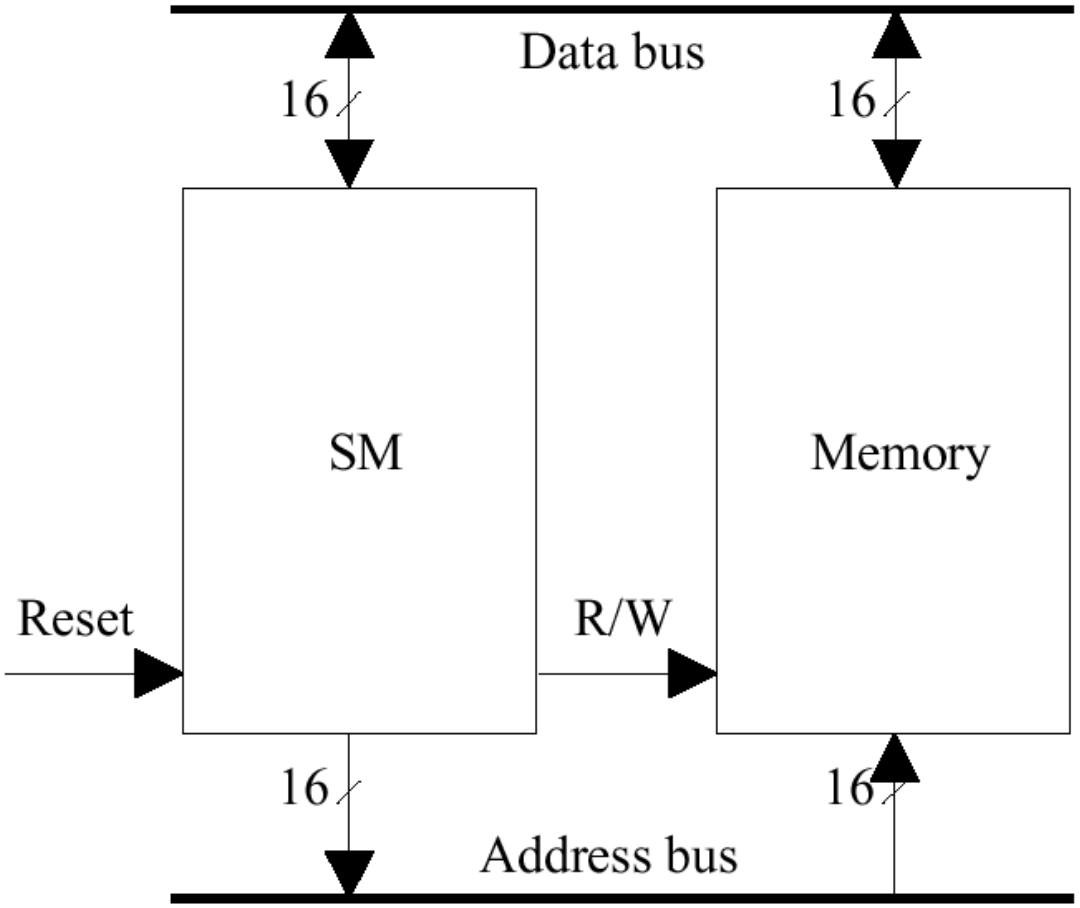

Fig. 8.4 Relationship between SM and a memory system.

SM Datapath

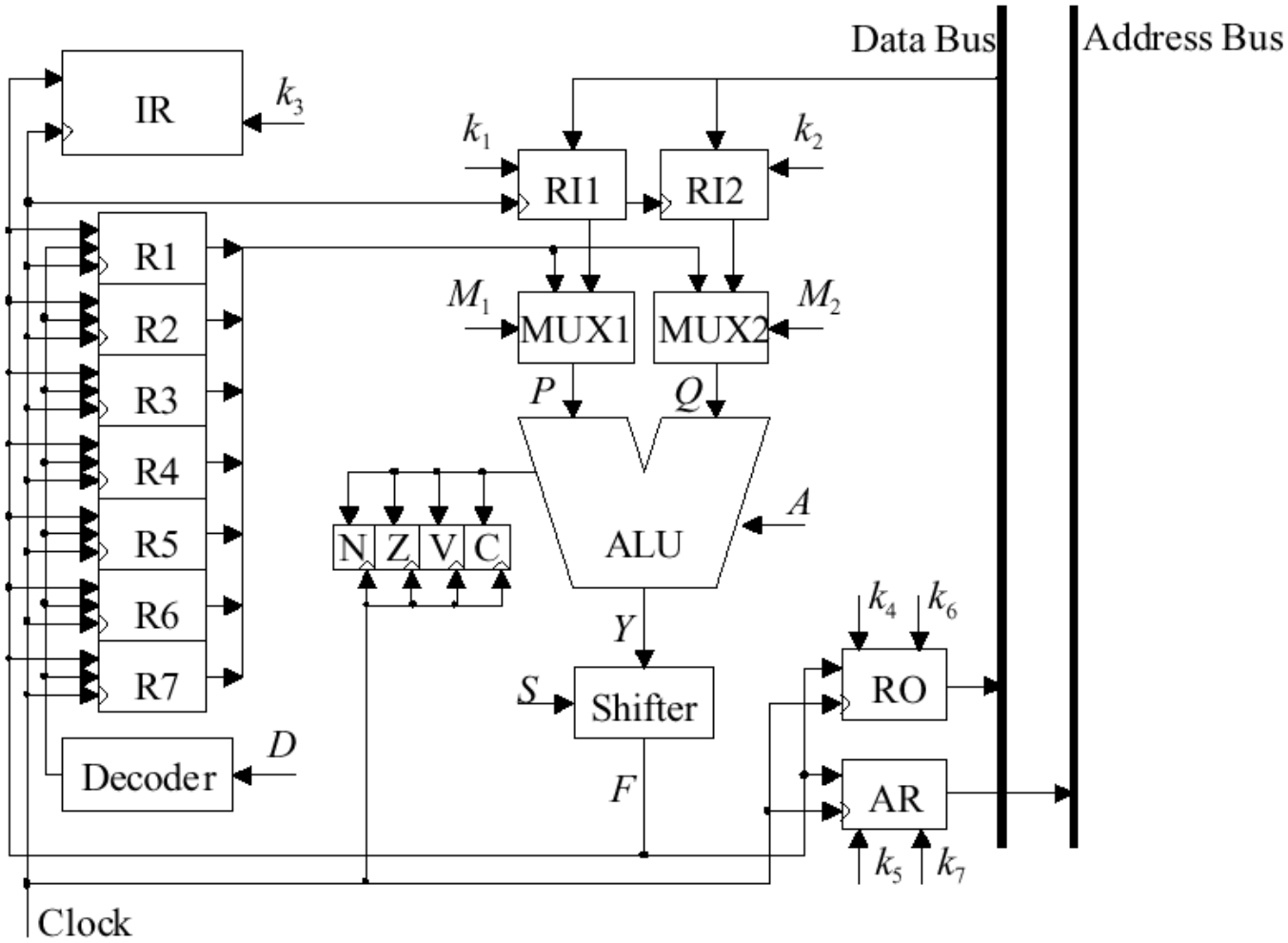

Fig. 8.5 Example SM datapath.

Control Signals

Code	A	M_1	M_2	S	D
000	$Y = P$	RI1 selected	RI2 selected	Bypass	
001	$Y = P + Q$	R1 selected	R1 selected	Logical Shift left[2]	R1 loaded
010	$Y = P + 1$	R2 selected	R2 selected	Logical Shift right	R2 loaded
011	$Y = P - Q$	R3 selected	R3 selected	Output = 0	R3 loaded
100	$Y = \overline{P}$	R4 selected	R4 selected	Arithmetic Shift Left	R4 loaded
101	$Y = P \wedge Q$	R5 selected	R5 selected	Arithmetic Shift Right	R5 loaded
110	$Y = P \vee Q$	R6 selected	R6 selected	Rotate Left	R6 loaded
111	$Y = P \oplus Q$	R7 selected	R7 selected	Rotate Right	R7 loaded

Fig. 8.6 Control signals for ALU, MUX1, MUX2, shifter, and decoder.

Datapath Operations

Operation	A	M_1	M_2	S	D	$k_1k_2k_3k_4k_5k_6k_7$
R1 ← R2 – R3	011	010	011	000	001	0000000
R2 ← Shift right (R5)	000	101	-	010	010	0000000
AR ← R7	000	111	-	000	000	0000100
Memory ← AR	-	-	-	-	000	0000001

Fig. 8.7 Datapath operation examples.

ALU Bit Slice

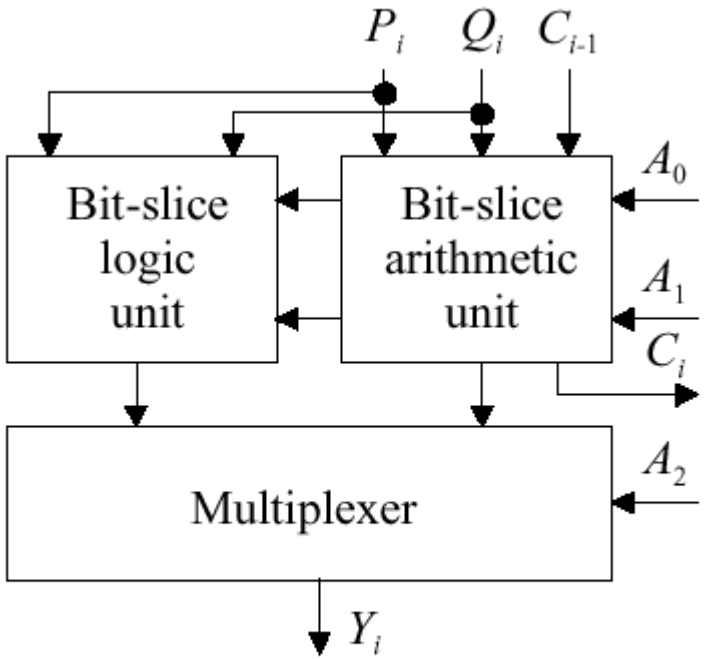

Fig. 8.8 A bit slice of the ALU.

Shift Register

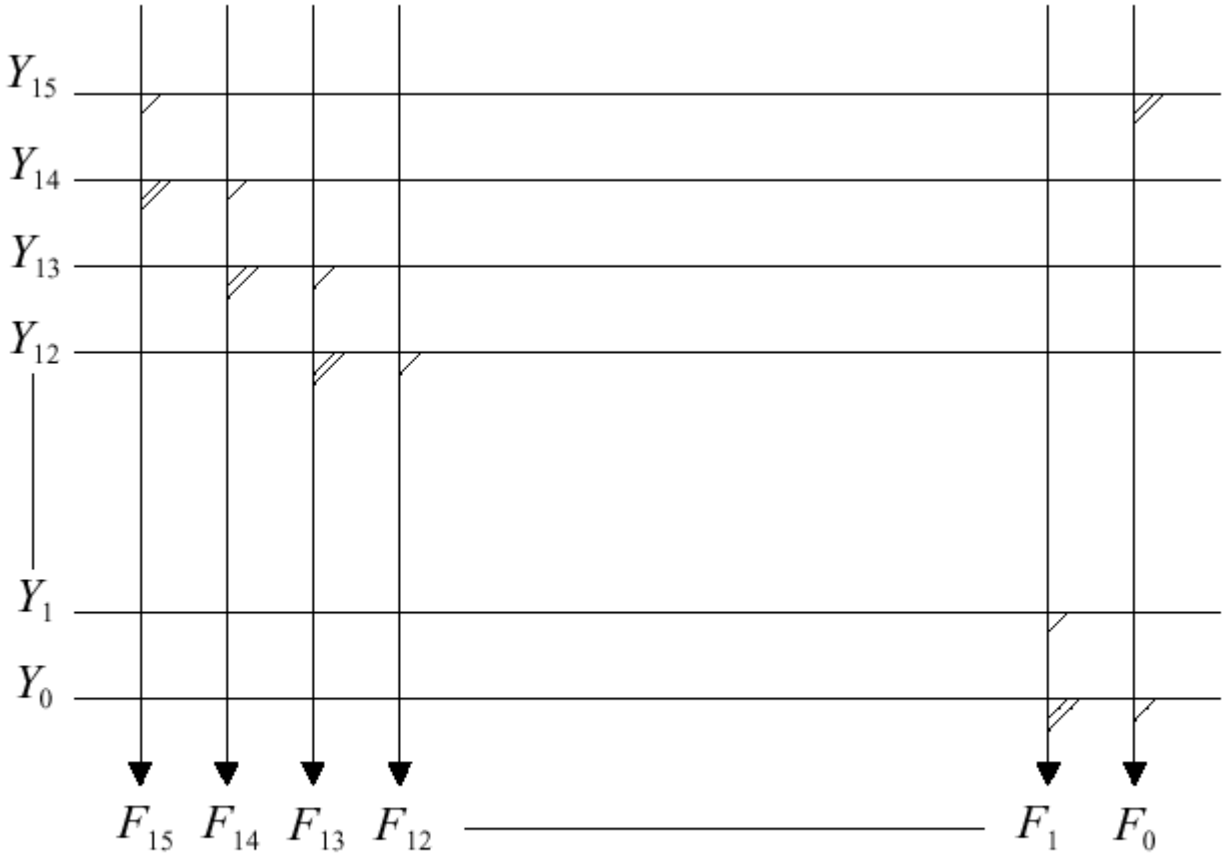

Fig. 8.9 Partial implementation of the shift register.

Instruction Format

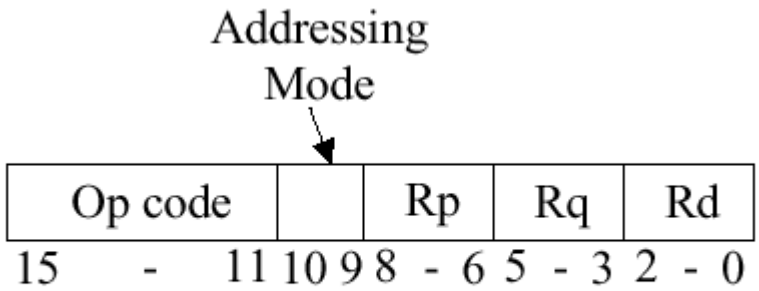

Fig. 8.10 Instruction format.

Addressing Modes

Bits 10, 9	Addressing Mode	Address Calculation
00	(Rx)	Address = Rx
01	*m*(Rx)	Address = *m* + Rx
10	(Rx)[Ri]	Address = Rx + Ri
11	constant	Address = PC + 1

Fig. 8.11 Addressing modes.

Control Unit

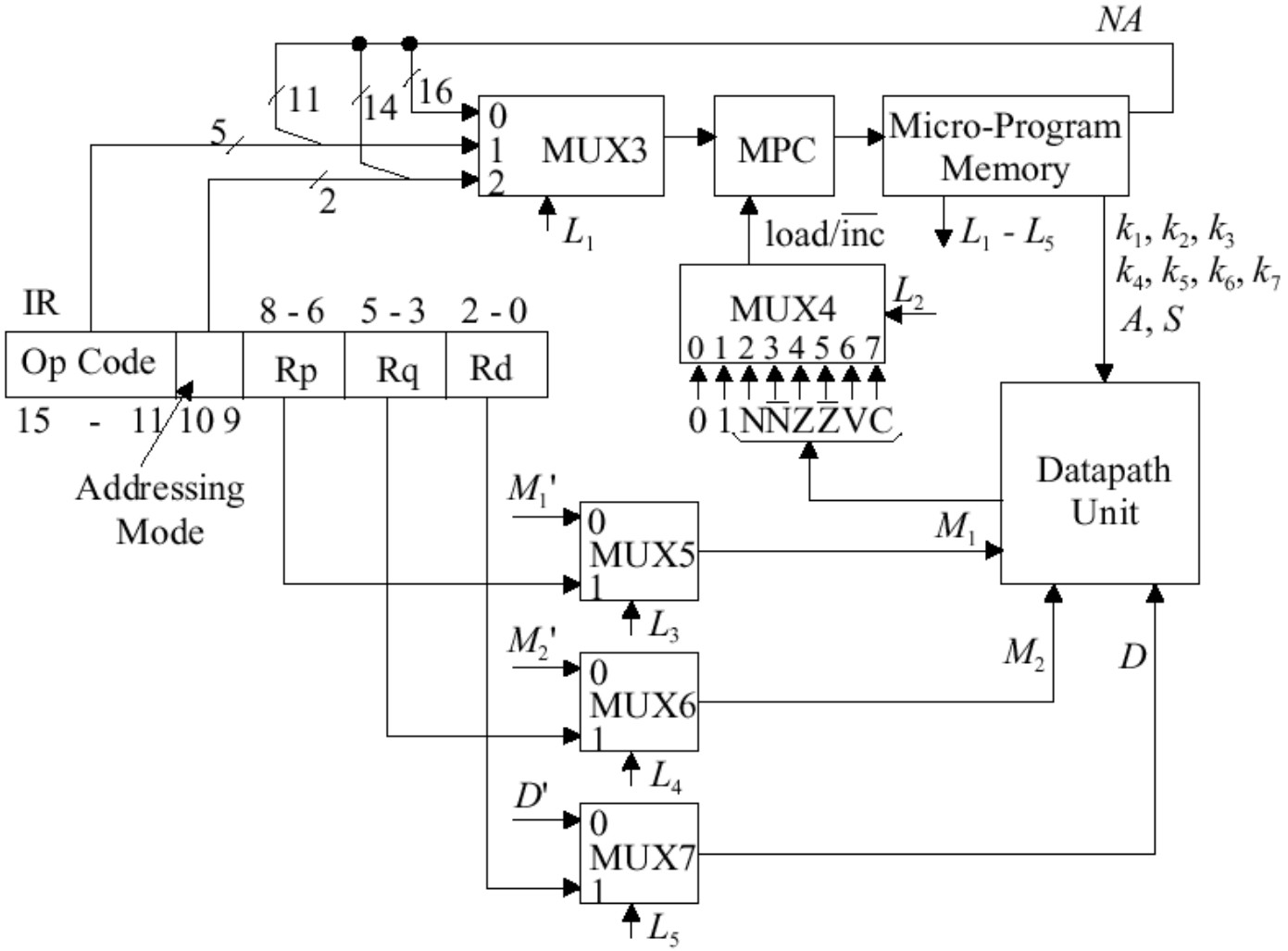

Fig. 8.12 Control unit of SM.

Microprogramming

	MPC Address	L_1	L_2	L_3	L_4	L_5	M_1'	M_2'	D'	NA	A	S	k_1-k_7
Instruction fetch:													
AR ← R7	0000	-	0	0	-	0	7	-	0	-	0	0	0000100
BUS ← AR; R7 ← R7 + 1	0001	-	0	0	-	0	7	-	7	-	2	0	0000001
RI1 ← BUS	0002	-	0	-	-	0	-	-	0	-	-	-	1000000
IR ← RI1	0003	-	0	0	-	0	0	-	0	-	0	0	0010000
Interpretation:													
determine next address	0004	1	1	-	-	0	-	-	0	0100	-	-	0000000
	....												
LOAD:													
addressing modes	0100	2	1	-	-	0	-	-	0	0200	-	-	0000000
(other instructions)…	....												
Mode 00:	0200	0	1	-	-	0	-	-	0	0204	-	-	0000000
Mode 01:	0201	0	1	-	-	0	-	-	0	0208	-	-	0000000
Mode 10:	0202	0	1	-	-	0	-	-	0	020C	-	-	0000000
Mode 11:	0203	0	1	-	-	0	-	-	0	020D	-	-	0000000
(Rx)													
AR ← Rx	0204	-	0	1	-	0	-	-	0	-	0	0	0000100
BUS ← AR	0205	-	0	-	-	0	-	-	0	-	-	-	0000001
RI1 ← BUS	0206	-	0	-	-	0	-	-	0	-	-	-	1000000
Rd ← RI1	0207	0	1	0	-	1	0	-	-	0000	0	0	0000000
m(Rx)													
AR ← R7	0208	-	0	0	-	0	7	-	0	-	0	0	0000100
BUS ← AR; R7 ← R7 + 1	0209	-	0	0	-	0	7	-	7	-	2	0	0000001
RI2 ← BUS	020A	-	0	-	-	0	-	-	0	-	-	-	0100000
AR ← RI2 + Rx	020B	0	1	1	0	0	-	0	0	0205	1	0	0000100
(Rx)[Ri]													
AR ← Ri + Rx	020C	0	1	1	1	0	-	-	0	0205	1	0	0000100
constant													
AR ← R7	020D	-	0	0	-	0	7	-	0	-	0	0	0000100
BUS ← AR; R7 ← R7 + 1	020E	0	1	0	-	0	7	-	7	0206	2	0	0000001
…	....												

Fig. 8.13 Microprogramming example.

FPGA Floorplan

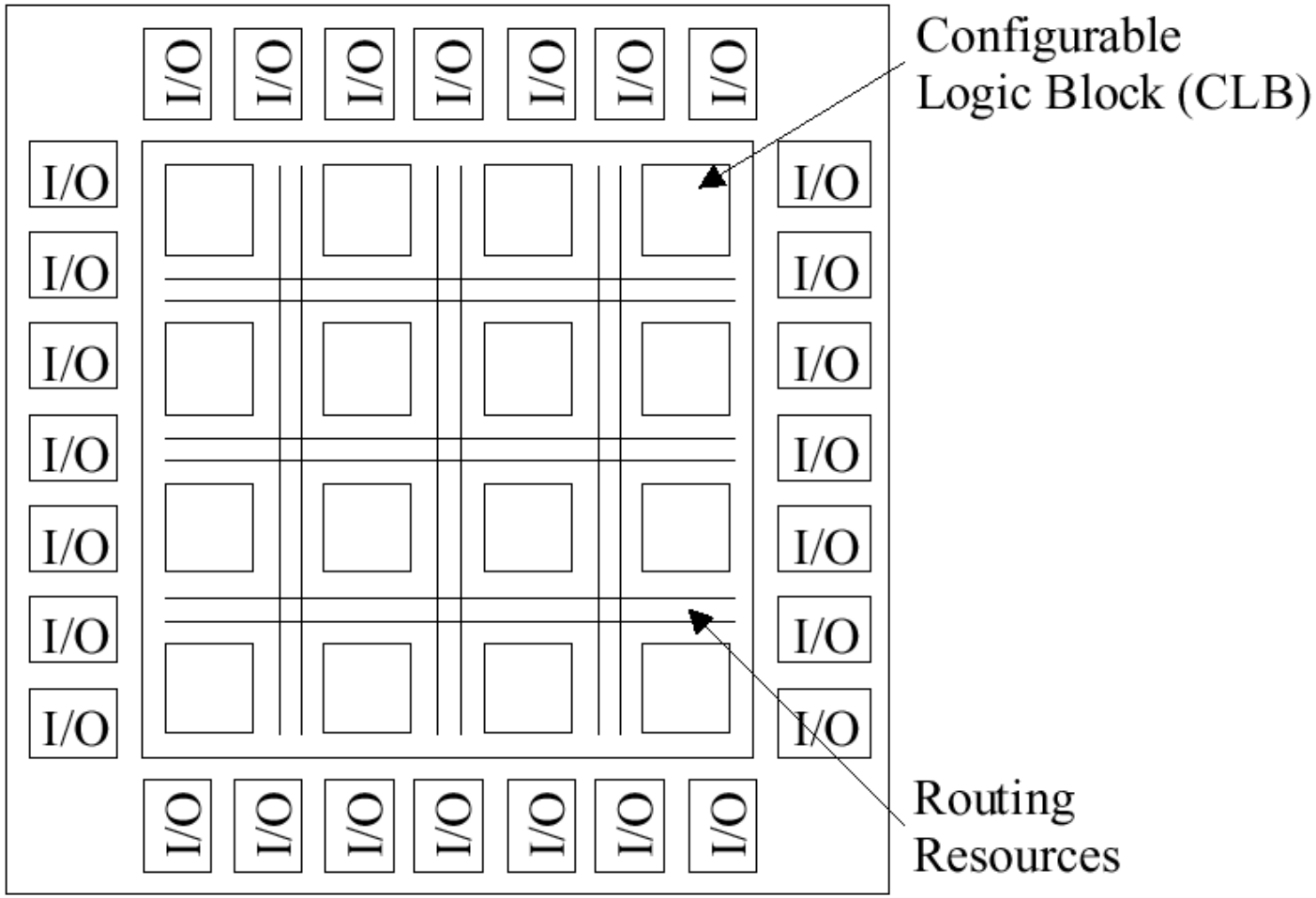

Fig. 8.14 General floorplan of an FPGA.

CLB

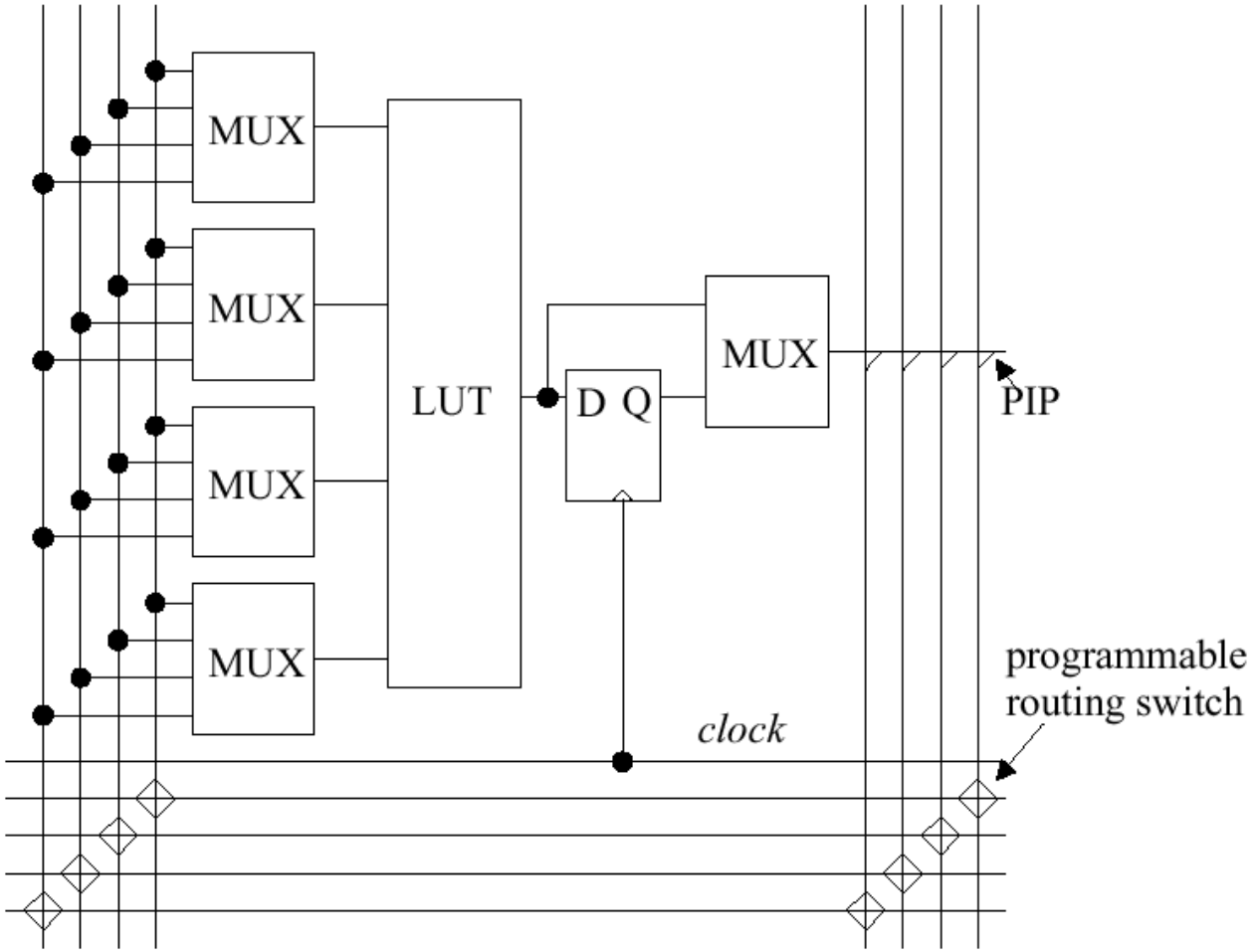

Fig. 8.15 Block diagram of CLB and its relationship with the routing resource.

Programmable Routing Switch

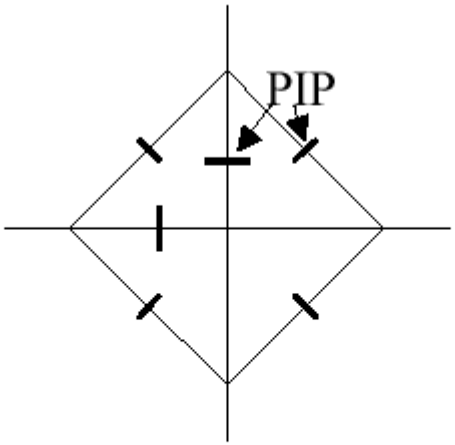

Fig. 8.16 Programmable routing switch.