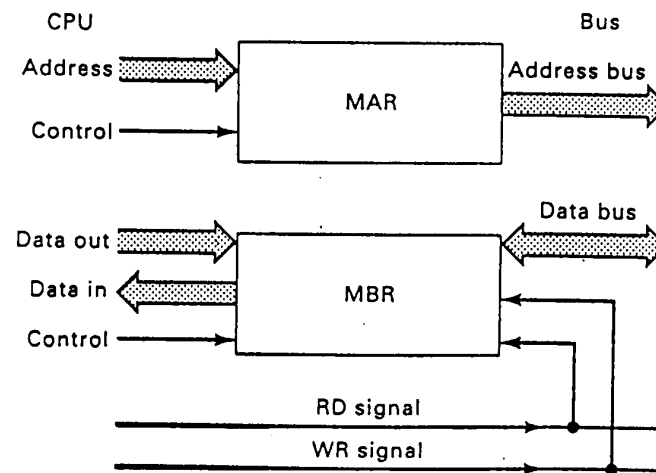
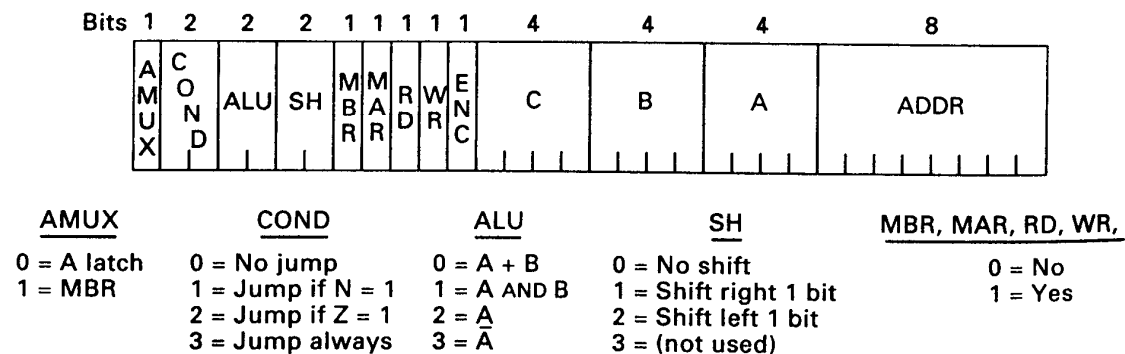


Fig. 4-5. (a) A clock with four outputs. (b) The output timing diagram.



AMUX — controls left ALU input: 0 = A latch, 1 = MBR  
 ALU — ALU function: 0 = A + B, 1 = A AND B, 2 = A, 3 =  $\bar{A}$   
 SH — shifter function: 0 = no shift, 1 = right, 2 = left  
 MBR — loads MBR from shifter: 0 = don't load MBR, 1 = load MBR  
 MAR — loads MAR from B latch: 0 = don't load MAR, 1 = load MAR  
 RD — requests memory read: 0 = no read, 1 = load MBR from memory  
 WR — requests memory write: 0 = no write, 1 = write MBR to memory  
 ENC — controls storing into scratchpad: 0 = don't store, 1 = store  
 C — selects register for storing into if ENC = 1: 0 = PC, 1 = AC, etc.  
 B — selects B bus source: 0 = PC, 1 = AC, etc.  
 A — selects A bus source: 0 = PC, 1 = AC, etc.



0: <i>mar</i> := <i>pc</i> ; <i>rd</i> ;	{main loop}
1: <i>pc</i> := <i>pc</i> + 1 ; <i>rd</i> ;	{increment <i>pc</i> }
2: <i>ir</i> := <i>mbr</i> ; if <i>n</i> then goto 28;	{save, decode <i>mbr</i> }
3: <i>tir</i> := <i>lshift</i> ( <i>ir</i> + <i>ir</i> ) ; if <i>n</i> then goto 19;	{000x or 001x?}
4: <i>tir</i> := <i>lshift</i> ( <i>tir</i> ) ; if <i>n</i> then goto 11;	{0000 or 0001?}
5: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 9;	{0000 = LODD}
6: <i>mar</i> := <i>ir</i> ; <i>rd</i> ;	
7: <i>rd</i> ;	
8: <i>ac</i> := <i>mbr</i> ; goto 0;	
9: <i>mar</i> := <i>ir</i> ; <i>mbr</i> := <i>ac</i> ; <i>wr</i> ;	{0001 = STOD}
10: <i>wr</i> ; goto 0;	
11: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 15;	{0010 or 0011?}
12: <i>mar</i> := <i>ir</i> ; <i>rd</i> ;	{0010 = ADDD}
13: <i>rd</i> ;	
14: <i>ac</i> := <i>mbr</i> + <i>ac</i> ; goto 0;	
15: <i>mar</i> := <i>ir</i> ; <i>rd</i> ;	{0011 = SUBD}
16: <i>ac</i> := <i>ac</i> + 1 ; <i>rd</i> ;	{Note: $x - y = x + 1 + \text{not } y$ }
17: <i>a</i> := <i>inv</i> ( <i>mbr</i> ) ;	
18: <i>ac</i> := <i>ac</i> + <i>a</i> ; goto 0;	
19: <i>tir</i> := <i>lshift</i> ( <i>tir</i> ) ; if <i>n</i> then goto 25;	{010x or 011x?}
20: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 23;	{0100 or 0101?}
21: <i>alu</i> := <i>ac</i> ; if <i>n</i> then goto 0;	{0100 = JPOS}
22: <i>pc</i> := <i>band</i> ( <i>ir</i> , <i>amask</i> ) ; goto 0;	{perform the jump}
23: <i>alu</i> := <i>ac</i> ; if <i>z</i> then goto 22;	{0101 = JZER}
24: goto 0;	{jump failed}
25: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 27;	{0110 or 0111?}
26: <i>pc</i> := <i>band</i> ( <i>ir</i> , <i>amask</i> ) ; goto 0;	{0110 = JUMP}
27: <i>ac</i> := <i>band</i> ( <i>ir</i> , <i>amask</i> ) ; goto 0;	{0111 = LOCO}
28: <i>tir</i> := <i>lshift</i> ( <i>ir</i> + <i>ir</i> ) ; if <i>n</i> then goto 40;	{10xx or 11xx?}
29: <i>tir</i> := <i>lshift</i> ( <i>tir</i> ) ; if <i>n</i> then goto 35;	{100x or 101x?}
30: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 33;	{1000 or 1001?}
31: <i>a</i> := <i>ir</i> + <i>sp</i> ;	{1000 = LODL}
32: <i>mar</i> := <i>a</i> ; <i>rd</i> ; goto 7;	
33: <i>a</i> := <i>ir</i> + <i>sp</i> ;	{1001 = STOL}
34: <i>mar</i> := <i>a</i> ; <i>mbr</i> := <i>ac</i> ; <i>wr</i> ; goto 10;	
35: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 38;	{1010 or 1011?}
36: <i>a</i> := <i>ir</i> + <i>sp</i> ;	{1010 = ADDL}
37: <i>mar</i> := <i>a</i> ; <i>rd</i> ; goto 13;	
38: <i>a</i> := <i>ir</i> + <i>sp</i> ;	{1011 = SUBL}
39: <i>mar</i> := <i>a</i> ; <i>rd</i> ; goto 16;	

40: <i>tir</i> := <i>lshift</i> ( <i>tir</i> ); if <i>n</i> then goto 46;	{110x or 111x?}
41: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 44;	{1100 or 1101?}
42: <i>alu</i> := <i>ac</i> ; if <i>n</i> then goto 22;	{1100 = JNEG}
43: goto 0;	
44: <i>alu</i> := <i>ac</i> ; if <i>z</i> then goto 0;	{1101 = JNZE}
45: <i>pc</i> := <i>band</i> ( <i>ir</i> , <i>amask</i> ); goto 0;	
46: <i>tir</i> := <i>lshift</i> ( <i>tir</i> ); if <i>n</i> then goto 50;	
47: <i>sp</i> := <i>sp</i> + (-1);	{1110 = CALL}
48: <i>mar</i> := <i>sp</i> ; <i>mbr</i> := <i>pc</i> ; <i>wr</i> ;	
49: <i>pc</i> := <i>band</i> ( <i>ir</i> , <i>amask</i> ); <i>wr</i> ; goto 0;	
50: <i>tir</i> := <i>lshift</i> ( <i>tir</i> ); if <i>n</i> then goto 65;	{1111, examine addr}
51: <i>tir</i> := <i>lshift</i> ( <i>tir</i> ); if <i>n</i> then goto 59;	
52: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 56;	
53: <i>mar</i> := <i>ac</i> ; <i>rd</i> ;	{1111000 = PSHI}
54: <i>sp</i> := <i>sp</i> + (-1); <i>rd</i> ;	
55: <i>mar</i> := <i>sp</i> ; <i>wr</i> ; goto 10;	
56: <i>mar</i> := <i>sp</i> ; <i>sp</i> := <i>sp</i> + 1; <i>rd</i> ;	{1111001 = POPI}
57: <i>rd</i> ;	
58: <i>mar</i> := <i>ac</i> ; <i>wr</i> ; goto 10;	
59: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 62;	
60: <i>sp</i> := <i>sp</i> + (-1);	{1111010 = PUSH}
61: <i>mar</i> := <i>sp</i> ; <i>mbr</i> := <i>ac</i> ; <i>wr</i> ; goto 10;	
62: <i>mar</i> := <i>sp</i> ; <i>sp</i> := <i>sp</i> + 1; <i>rd</i> ;	{1111011 = POP}
63: <i>rd</i> ;	
64: <i>ac</i> := <i>mbr</i> ; goto 0;	
65: <i>tir</i> := <i>lshift</i> ( <i>tir</i> ); if <i>n</i> then goto 73;	
66: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 70;	
67: <i>mar</i> := <i>sp</i> ; <i>sp</i> := <i>sp</i> + 1; <i>rd</i> ;	{1111100 = RETN}
68: <i>rd</i> ;	
69: <i>pc</i> := <i>mbr</i> ; goto 0;	
70: <i>a</i> := <i>ac</i> ;	{1111101 = SWAP}
71: <i>ac</i> := <i>sp</i> ;	
72: <i>sp</i> := <i>a</i> ; goto 0;	
73: <i>alu</i> := <i>tir</i> ; if <i>n</i> then goto 76;	
74: <i>a</i> := <i>band</i> ( <i>ir</i> , <i>smask</i> );	{1111110 = INSP}
75: <i>sp</i> := <i>sp</i> + <i>a</i> ; goto 0;	
76: <i>a</i> := <i>band</i> ( <i>ir</i> , <i>smask</i> );	{1111111 = DESP}
77: <i>a</i> := <i>inv</i> ( <i>a</i> );	
78: <i>a</i> := <i>a</i> + 1; goto 75;	