Exercise 7. (a)

The folding delays and folding constraints are shown in Table 6.2. The constraint graph is shown in Fig. 6.8. Using Bellman-Ford algorithm, the retiming values are givens as

$$r(A_1) = -2, r(A_2) = -3, r(A_3) = 0, r(A_4) = 0,$$
$$r(M_1) = -4, r(M_2) = -3, r(M_3) = -1, r(M_4) = -2, r(M_5) = -1.$$

The retimed SFG is shown in Fig. 6.9. The folding delays are recalcu-

Folding delays	Folding constraints
$D_F(A_2 \longrightarrow M_2) = 2 \times 0 - 1 + 1 - 0 = 0$	$r(A_2) - r(M_2) \leq 0$
$D_F(A_2 \longrightarrow A_1) = 2 \times 2 - 1 + 1 - 0 = 4$	$r(A_2) - r(A_1) \leq 2$
$D_F(A_2 \longrightarrow M_1) = 2 \times 2 - 1 + 0 - 0 = 3$	$r(A_2) - r(M_1) \leq 1$
$D_F(A_2 \longrightarrow M_4) = 2 \times 1 - 1 + 1 - 0 = 2$	$r(A_2) - r(M_4) \leq 1$
$D_F(A_2 \longrightarrow M_5) = 2 \times 0 - 1 + 1 - 0 = 0$	$r(A_2) - r(M_5) \leq 0$
$D_F(M_2 \longrightarrow A_1) = 2 \times 0 - 2 + 1 - 1 = -2$	$r(M_2) - r(A_1) \leq -1$
$D_F(M_1 \longrightarrow A_2) = 2 \times 0 - 2 + 0 - 0 = -2$	$r(M_1) - r(A_2) \leq -1$
$D_F(A_1 \longrightarrow M_3) = 2 \times 0 - 1 + 0 - 1 = -2$	$r(A_1) - r(M_3) \leq -1$
$D_F(M_3 \longrightarrow A_3) = 2 \times 0 - 2 + 0 - 0 = -2$	$r(M_3) - r(A_3) \leq -1$
$D_F(M_4 \longrightarrow A_3) = 2 \times 0 - 2 + 0 - 1 = -3$	$r(M_4) - r(A_3) \leq -2$
$D_F(M_5 \longrightarrow A_4) = 2 \times 0 - 2 + 1 - 1 = -2$	$r(M_5) - r(A_4) \leq -1$
$D_F(A_3 \longrightarrow A_4) = 2 \times 0 - 1 + 1 - 0 = 0$	$r(A_3) - r(A_4) \leq 0$

Table 6.2 The folding delay and constraint table for problem 7(a).

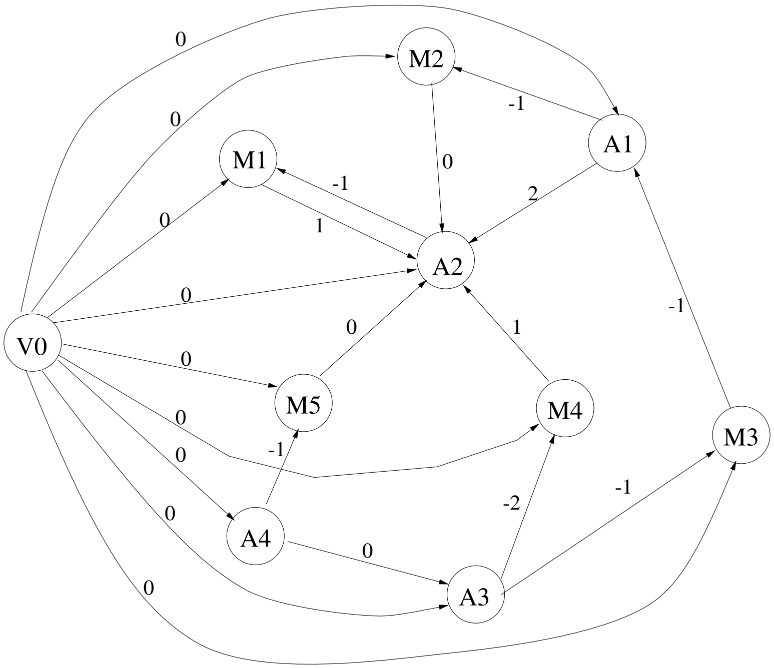

Fig. 6.8 The constrained graph for problem 7(a).

lated as

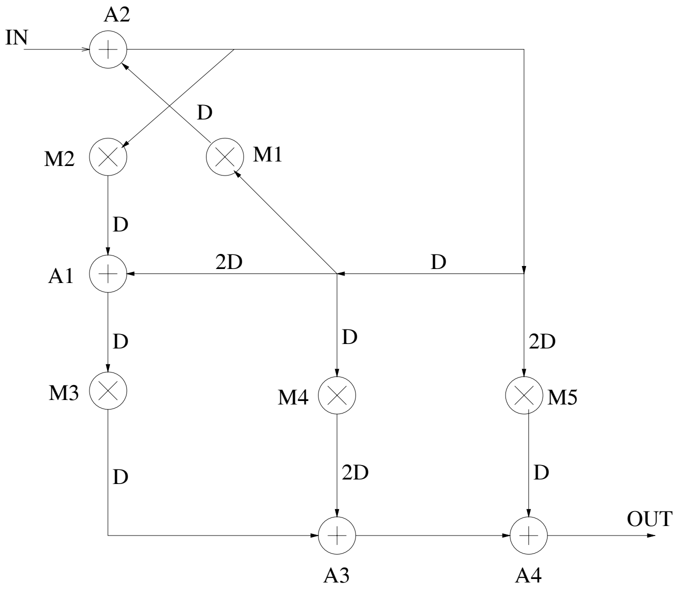

Fig. 6.9 The retimed SFG for problem 7(a).

$$
\begin{aligned}
D_F(A_2 &\longrightarrow M_2) = 2\times 0 - 1 + 1 - 0 = 0\\
D_F(A_2 &\longrightarrow A_1) = 2\times 3 - 1 + 1 - 0 = 6\\
D_F(A_2 &\longrightarrow M_1) = 2\times 1 - 1 + 0 - 0 = 1\\
D_F(A_2 &\longrightarrow M_4) = 2\times 2 - 1 + 1 - 0 = 4\\
D_F(A_2 &\longrightarrow M_5) = 2\times 2 - 1 + 1 - 0 = 4\\
D_F(M_2 &\longrightarrow A_1) = 2\times 1 - 2 + 1 - 1 = 0\\
D_F(M_1 &\longrightarrow A_2) = 2\times 1 - 2 + 0 - 0 = 0\\
D_F(A_1 &\longrightarrow M_3) = 2\times 1 - 1 + 0 - 1 = 0\\
D_F(M_3 &\longrightarrow A_3) = 2\times 1 - 2 + 0 - 0 = 0\\
D_F(M_4 &\longrightarrow A_3) = 2\times 2 - 2 + 0 - 1 = 1\\
D_F(M_5 &\longrightarrow A_4) = 2\times 1 - 2 + 1 - 1 = 0\\
D_F(A_3 &\longrightarrow A_4) = 2\times 0 - 1 + 1 - 0 = 0
\end{aligned}
$$

(b) The folded architecture is shown in Fig. 6.10.

Exercise 8. [Solution]:

The life time table is shown in Table 6.3. The latency is 7 unit time. The life time chart is shown in Fig. 6.11, and the minimum number of registers is 8. The allocation table using forward-backward allocation

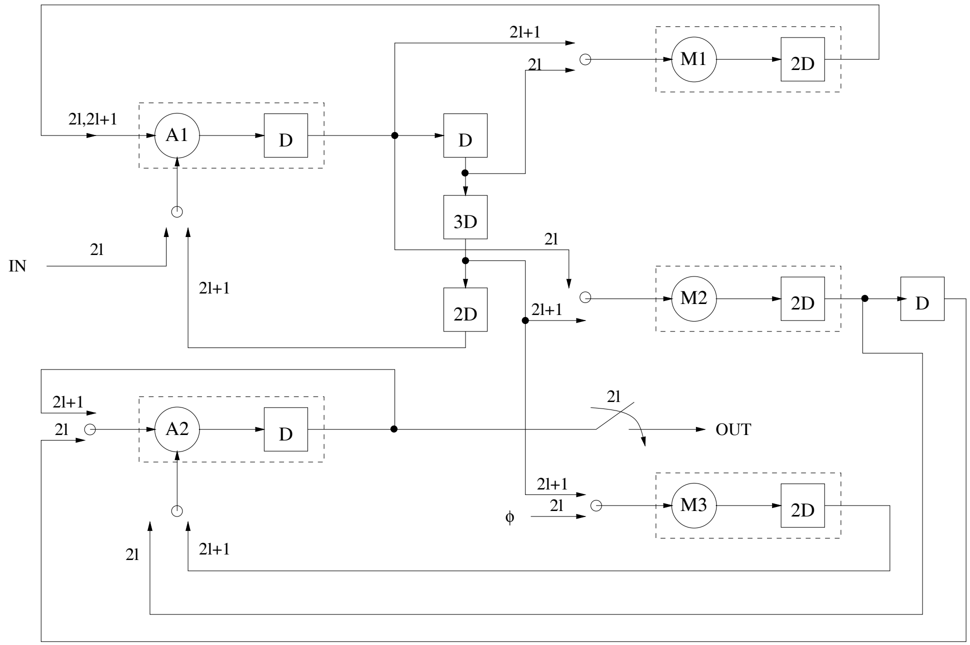

Fig. 6.10 The folded architecture for problem 7(b).

Variable	T_{input}	T_{output}	T_{diff}	T_{out}	Duration
x_0	0	0	0	7	7
x_1	0	6	6	13	13
x_2	2	0	−2	7	5
x_3	2	6	4	13	11
x_4	4	0	−4	7	3
x_5	4	6	2	13	9
x_6	6	3	−3	10	4
x_7	6	9	3	16	10
x_8	8	3	−5	10	2
x_9	8	9	1	16	8
x_{10}	10	3	−7	10	0
x_{11}	10	9	−1	16	6

Table 6.3 The life time table for problem 8.

scheme is shown in Fig. 6.12. The synthesized architecture is shown in Fig. 6.13

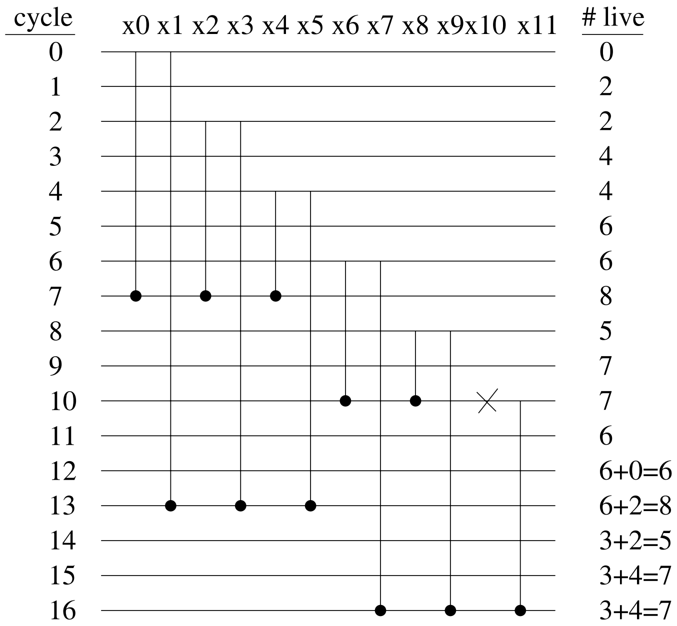

Fig. 6.11 The life time chart for problem 8.

cycle	input	R1	R2	R3	R4	R5	R6	R7	R8	output
0	**x0 x1**									
1		**x1**	**x0**							
2	**x2 x3**		**x1**	**x0**						
3		**x3**	**x2**	**x1**	**x0**					
4	**x4 x5**		**x3**	**x2**	**x1**	**x0**				
5		**x5**	**x4**	**x3**	**x2**	**x1**	**x0**			
6	**x6 x7**		**x5**	**x4**	**x3**	**x2**	**x1**	**x0**		
7		**x7**	**x6**	**x5**	**x4**	**x3**	**x2**	**x1**	**x0**	**x0 x2 x4**
8	**x8 x9**		**x7**	**x6**	**x5**		**x3**		**x1**	
9		**x9**	**x8**	**x7**	**x6**	**x5**	**x1**	**x3**		
10	**x10 x11**		**x9**	**x8**	**x7**	**x6**	**x5**	**x1**	**x3**	**x6 x8 x10**
11		**x11**		**x9**		**x7**	**x3**	**x5**	**x1**	
12			**x11**		**x9**	**x1**	**x7**	**x3**	**x5**	
13				**x11**	**x5**	**x9**	**x1**	**x7**	**x3**	**x1 x3 x5**
14					**x11**		**x9**		**x7**	
15						**x11**	**x7**	**x9**		
16							**x11**	**x7**	**x9**	**x7 x9 x11**

Fig. 6.12 The allocation table using forward-backward method for problem 8.

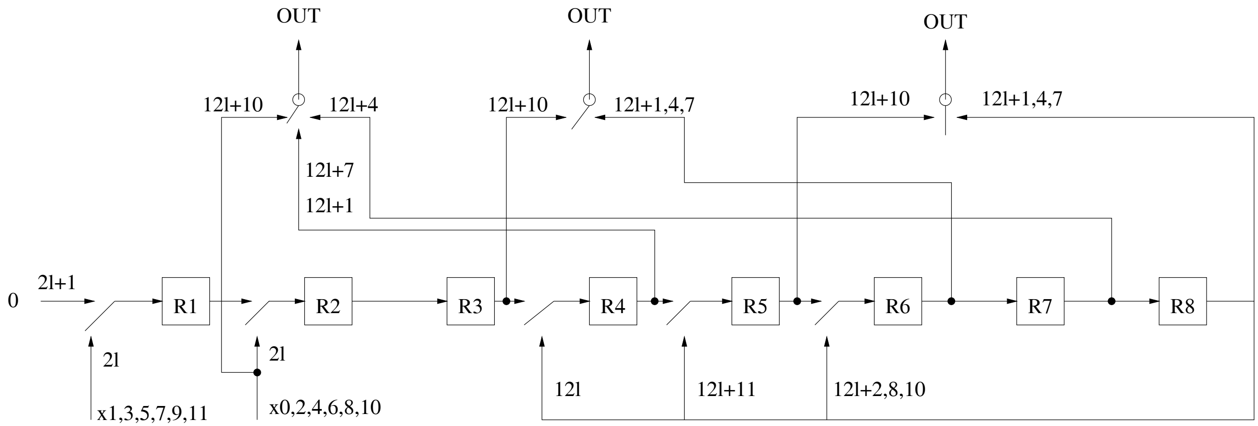

Fig. 6.13 The synthesized architecture for problem 8.