Proceedings of the 7th Biennial Baltic Electronic Conference BEC'2000

FSM Decomposition Using Shift Registers

M. Kruus , H. Lensen

Department of Computer Engineering, TTU, Raja 15, Tallinn, Estonia,
e-mail: kruus@cc.ttu.ee hl@cc.ttu.ee

ABSTRACT: In this work we present a method of Finite State Machine (FSM) decomposition where shift registers are used as memory of FSM network components. Every component of the network is realized on a separate shift register. This approach improves the testability of the FSM network. State splitting method is described for the minimization of the number of used shift registers. Described algorithms are illustrated by examples. As conclusion, the overview of the experimental results is presented.

1 Introduction

FSM (Finite State Machine) testing has been always a complicated problem. One way to solve the testing problem is based on Checking Experiments Theory [2]. As an improvement to this approach, the length of the CS can be significantly reduced by using shift registers as FSM memory [1]. This solution reduces the complexity of the test experiments. If a FSM is realized on the shift register with length N and register's content is directly visible via output function, then any sequence of N input symbols appears to be Distinguishing [2] for this particular FSM.

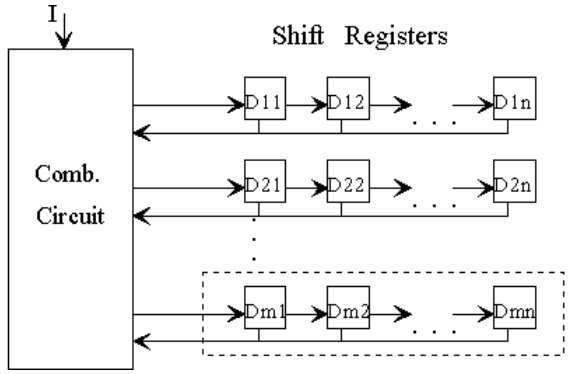

Fig. 1

Intention to realize a FSM on the shift registers (*Fig. 1*) may be treated as a special decomposition problem of this initial FSM [3]. Such a decomposition can be accomplished using a complete set of partitions
$P = \{ \pi_i \mid 1 \le i \le n \}$
where each partition of states consists of 2 blocks. Each partition in set P corresponds to one position in shift register. In [3] the conditions, required for the existence of such a decomposition are listed. In our current work we describe the algorithms for FSM realization on shift registers, using the state splitting method of the initial FSM.

2 Basic Notations

FSM model as our research target may be treated as triple $A = (I , S , \delta)$ or in other words — the output function of the automation is dropped as unimportant in our context. In this triple:
S – set of automation's states;
I – input alphabet;
$\delta: S \times I \rightarrow S$ — transition function of FSM

Definition 1: *Partition* on the set S is a set of disjoint subsets: $\{ B_1 B_2 \ldots B_n \}$ where each $B_i \subset S$;
$B_1 \cup B_2 \cup \ldots \cup B_n = S$;
$B_i \cap B_j = \varnothing$
$1 \le i, j \le n; \quad i \ne j$

Definition 2: Set of partitions $\{ \pi_1 \ \pi_2 \ \ldots \ \pi_n \}$ is *complete*, if $\pi_1 \cdot \pi_2 \cdot \ldots \cdot \pi_n = 0_\pi$

Definition 3: $\pi [s] = B$, where $B \in \pi$ and $s \in B$

Definition 4: *Partition pair* (π , π') is called *symmetric* if there exists one to one mapping $\varphi : \pi \leftrightarrow \pi'$ so that for any block $B \in \pi$
$\varphi (B) = \pi' [\delta (s, a)]$ where
$B \in \pi \quad s \in B \quad a \in I$

Definition 5: N-chain is a sequence of symmetric partition pairs (π_{i-1} , π_i) where $2 \le i \le N$

The existence of this N-chain is the required and sufficient condition, indicating that the considered automation may be realized on shift register with length N [3]. If the set of partitions is *complete*, then the FSM can be realized on a single shift register.

Let us introduce the set of all possible symmetric partition pairs
$\{ (\pi_j , \pi_j') \}$ of a FSM.

$\pi^c = \prod \pi_j$ and $\pi^r = \prod \pi_j'$ are the least partitions which build up a symmetric partition pair.

In [3] are described the algorithms finding these partitions π^c and π^r and all N-chains for a particular FSM. Denote that N-chains constructed in [3] does not

consist of 2-block partitions. Each N-chain describes in [3] a FSM network, where components $A_1 \ldots A_n$ have more than 2 states. In other words, the realization of $A_1 \ldots A_n$ requires $] \log_2 |\pi_i| [$ shift registers with length N. $|\pi_i|$ is the number of blocks in partition π_i and expression $] e [$ forces the nearest integer value above e.

3 Realization of FSM on shift registers

In [3] it remains unspecified, how to choose from the set of all N-chains a minimal subset for realization of FSM on shift registers. In the following we describe a selection algorithm, allowing to minimize the total summary length of used shift registers.

Let us assume, that some N-chain has t blocks in each partition. From such a N-chain can be derived $(2^{t-1} - 1)$ different N-chains, where each partition contains 2 blocks. Further we call them *2-block N-chains*. Each N-chain corresponds in realization to one shift register with length N.

Each 2-block N-chain describes some subautomation of the initial FSM. For evaluation and comparision of the different available N-chains it is reasonable to introduce *quantitative evaluation function*. This function uses the multiplication π_t of all partitions in a N-chain:

$$\pi_t = \prod_{i=1}^{N} \pi_i \qquad (1)$$

It's evident, that $\pi_t \geq \pi^c \cdot \pi^r$ and different partitions π_t may be relatively to each other incomparable. The evaluation function for N-chains [4]:

$$R(\pi_t) = M \cdot E(\pi_t) + (M - |\pi_t|) \qquad (2)$$

where

M — number of FSM states in partition π_t

$E(\pi_t)$ — number of states in the greatest block of π_t.

$|\pi_t|$ — number of blocks in π_t.

If $\pi_t = 0_\pi$ then $R(\pi_t) = M$. If $\pi_t = 1_\pi$ then $R(\pi_t) = M^2 + M - 1$. Consequently, N-chain realizes the more of automation, the less the value of function $R(\pi_t)$ is.

The problem of FSM realization on shift registers turns to problem of finding the minimal set of 2-block N-chains, where $\Pi \pi_t = \pi^c \cdot \pi^r$.

In the following we describe the algorithm, composing the complete set of partitions π_i:

$\Pi \pi_i = 0_\pi$, appropriate for shift register realization of FSM.

Algorithm 1

1. Find all N-chains [3].
2. Compose all 2-block N-chains.
3. For each N-chain calculate the multiplication of its partitions π_t and evaluation function $R(\pi_t)$ (2).
4. Assign $\tau = 1_\pi$
5. Find $\min_t \{ R(\tau \cdot \pi_t) \}$.

N-chain, corresponding to the multiplication π_t, must be added to the set of partitions under construction. If there are N-chains having the equal value of evaluation function, then the shorter chain is preferred and will be chosen.

6. Assign $\tau = \tau \cdot \pi_t$
7. if $\tau = \pi^c \cdot \pi^r$ then go to step 10.
8. if $E(\tau) = 2$ then go to step 11.
9. go to step 5.
10. if $\pi^c \cdot \pi^r = 0_\pi$ then go to step 12.
11. if for partitions $\{\pi_i\}$ constructed so far $\Pi \pi_i > 0_\pi$, then add 1 appropriate partition to ensure $\Pi \pi_i = 0_\pi$.
12. End.

Condition $E(\tau) = 2$ on step 8 has the following explanation. If $E(\tau) = 2$, then the set of partitions $\{\pi_i\}$ may be forced to meet $\Pi \pi_i = 0_\pi$ by adding just one 2-block partition. In other words, we expand the shift register by one bit. It's evident, that this is the best solution in our case.

Example 1

Let us have the following automation A (*Table 1*).

s \ I	a	b		s \ I	**a**	**b**
1	2	3		**13**	14	14
2	3	4		**14**	15	15
3	5	5		**15**	16	16
4	6	6		**16**	17	17
5	4	4		**17**	18	18
6	7	7		**18**	1	1
7	8	10		**19**	20	20
8	9	9		**20**	21	21
9	10	12		**21**	18	22
10	11	11		**22**	23	23
11	12	19		**23**	22	18
12	13	13				

Table 1. FSM A

FSM A has 2-, 3-, 4-, 5- and 6-chains.

Applying algorithm described above and minimal values of the evaluation function, we get the first 2-chain (π_1, π_2) with following partitions:

$\pi_1 = \{ \{ 4, 7, 8, 9, 11, 12, 14, 17, 20, 21, 22, 23 \}$
$\{ 1, 2, 3, 5, 6, 10, 13, 15, 16, 18, 19 \} \};$

$\pi_2 = \{ \{ 6, 8, 9, 10, 12, 13, 15, 18, 19, 21, 22, 23 \}$
$\{ 1, 2, 3, 4, 5, 7, 11, 14, 16, 17, 20 \} \};$

Second pass of the algorithm collects also 2-chain (π_3, π_4) where:

$\pi_3 = \{ \{ 3, 4, 6, 8, 15, 16, 17, 18, 19, 21, 22, 23 \}$
$\{ 1, 2, 5, 7, 9, 10, 11, 12, 13, 14, 20 \} \};$

$\pi_4 = \{ \{ 1, 5, 6, 7, 9, 16, 17, 18, 20, 22, 23 \}$
$\{ 2, 3, 4, 8, 10, 11, 12, 13, 14, 15, 19, 21 \} \};$

Because the greatest block of the multiplication $\pi_1 \cdot \pi_2 \cdot \pi_3 \cdot \pi_4$ contains 2 states, we need one additional partition to split this block also. Suitable partition could be:

$\pi_5 = \{ \{ 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 15, 22 \}$
$\{ 5, 13, 14, 16, 17, 18, 19, 20, 21, 23 \} \};$

Consequently, the treated FSM A can be realized using two 2-bit shift registers and one additional trigger (*Fig. 2*). Denote, that the minimal number of the single memory elements (triggers) needed for the realization of the same FSM is also 5.

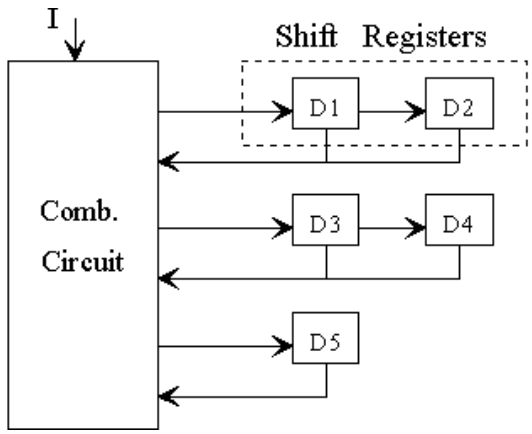

Fig. 2

4 State splitting of the initial FSM

However, it appears to be impossible completely to realize any random FSM on shift registers. In this case another approach can be used, where original FSM is altered to another functionally equivalent automation, using *state splitting method*. Obtained in this way modified FSM is in turn suitable for shift register realization.

Definition 6: *Cover* on the set S is a set of subsets:
$\{ B_1 B_2 \ldots B_n \}$ where each $B_i \subset S$;
$B_1 \cup B_2 \cup \ldots \cup B_n = S$; but
$B_i \cap B_j \neq \varnothing$ for some blocks B.
$1 \le i, j \le n; \quad i \neq j$

Let us have a partition of S marked π and a cover of S marked φ.

Definition 7: Pair (π, φ) is called *symmetric mixed pair*, if between blocks of π and φ can be established one-to-one mapping $\psi : \pi \leftrightarrow \varphi$ so that for each $B \in \pi$
$\psi(B) = \varphi [\delta (s,a)]$
where $B \in \pi$, $s \in B$, $a \in I$

Let us assume, that symmetric partition pairs (π_{i-1}, π_i) $2 \le i \le N$ build an N-chain and (π_N, φ) is a symmetric mixed pair.

Definition 8: Chain $(\pi_1, \pi_2)(\pi_2, \pi_3) \ldots .$ $(\pi_{N-1}, \pi_N)(\pi_N, \varphi)$ is called *extended N-chain*.

Let us assume $\pi_N = \{ B_1, B_2 \}$. Appropriate φ giving symmetric mixed pair with π_N can be found in following way:

$\varphi = \delta(\pi_N) = \{ \delta(B_1), \delta(B_2) \}$ where
$\delta(B_i) = \{ \cup \delta(s, x) \mid s \in B_i, x \in I \}$

Splitting those states of the initial FSM, which reside simultaneously in both blocks of φ, we obtain another modified FSM with set of states S'. For this new one there exists (N+1)-chain on the set of splitted states.
In following we present an algorithm for composing 2-block (N+1)-chain on the splitted set of states S', giving the corresponding automation on this set S'.

Algorithm 2

1. $\varphi = \delta(\pi_N)$. Build extended N-chain.
2. Find the set of splitted states:
   $v = \delta(B_1) \cap \delta(B_2)$
3. For each state s_i compose a related set $D(s_i)$ in following way:
$$D(s_i) = \begin{cases} \{ s_i \mid s_i \notin v \} \\ \{ s_i^1, s_i^2 \mid s_i \in v \} \end{cases}$$
4. Create the set of states of modified FSM:
   $S' = \bigtimes_{s_i \in S} D(s_i)$
5. In partitions π_j, $1 \le j \le N$ substitute the state $s_i \in v$ to its related set $D(s_i)$.
6. Turn the set φ into partition π_{N+1} in following way:

If $s_i \in \delta(B_1)$ & $s_i \in v$ then replace s_i with s_i^1;
If $s_i \in \delta(B_2)$ & $s_i \in v$ then replace s_i with s_i^2.

7. Define the new transition function δ, resulting the partitions π_N and π_{N+1} appear to be a symmetric partition pair on the set S'.
8. End

Example 2

$s \setminus I$	b	b
1	1	5
2	4	2
3	5	5
4	6	3
5	2	4
6	2	2

Table 2. FSM B

$s \setminus I$	a	b
1	1	5
$\mathbf{2^1}$	4^2	2^2
$\mathbf{2^2}$	4^2	2^2
3	5	5
$\mathbf{4^1}$	6	3
$\mathbf{4^2}$	6	3
5	2^1	4^1
6	2^1	2^1

Table 3. FSM B'

Let us assume that for automation B (*Table 2*) the following partitions π^c and π^r are calculated:

$\pi^c = \{ \{1, 3\} \{2, 5, 6\} \{4\} \}$

$\pi^r = \{ \{1, 5\} \{2, 4\} \{3, 6\} \}$

There are no 3-chains for automation B. Using minimal value of evaluation function $R(\pi_t)$ we choose 2-block 2-chain, containing following partitions π_1 and π_2:

$\pi_1 = \{ \{ 1, 3, 4 \} \{ 2, 5, 6 \} \}$

$\pi_2 = \{ \{ 1, 3, 5, 6 \} \{ 2, 4 \} \}$

At this point algorithm 2 is applied. We obtain

$\varphi = \delta(\pi_2) = \{ \{1, 2, 4, 5\} \{2, 3, 4, 6\} \}$

$v = \{1, 2, 4, 5\} \cap \{2, 3, 4, 6\} = \{2, 4\}$

$D(2) = \{ 2^1, 2^2 \}$ $D(4) = \{ 4^1, 4^2 \}$

Set of splitted states contains 8 members:

$S' = \{1, 2^1, 2^2, 3, 4^1, 4^2, 5, 6 \}$

Algorithm steps 5 and 6 compose 3-chain on the set S' :

$\pi_1 = \{ \{ 1, 3, 4^1, 4^2 \} \{ 2^1, 2^2, 5, 6 \} \}$

$\pi_2 = \{ \{ 1, 3, 5, 6 \} \{ 2^1, 2^2, 4^1, 4^2 \} \}$

$\pi_3 = \{ \{ 1, 2^1, 4^1, 5 \} \{ 2^2, 3, 4^2, 6 \} \}$

In algorithm step 7 we achieve a splitted FSM B' (*Table 3*).

Due to $\pi_1 \cdot \pi_2 \cdot \pi_3 = 0_\pi$ the FSM B' can be realized entirely using one 3-bit shift register.

State splitting method may be applied iteratively, i.e. the obtained splitted automation may be again treated as an initial FSM and algorithm 2 can be used repeatedly.

5 Conclusions

??????? ???? ???? ???? ???? ???? ????? ???? ???

Described state splitting method significantly increases the range of automations, having shift register realizations.

References

[1] Fuijawara H., Kinoshita K. „Design of sequential machines utilizing extra outputs". *IEEE Trans. Comput*. Vol. C-23, N 2, pp. 138-145, 1974.

[2] A.Gill, *Introduction to the Theory of Finite State Machines*, Mc-Graw Hill Book Co, 1962.

[3] Keevallik A., Leis P., Kruus M. "MPA realization on shift registers". *Publications of TPI.* Nr. 577, pp.85-95, 1984. (*in russian*)

[4] O'Keefe K.H., Johnson D.L. „The application of shift registers to secondary state assignment". Part I, II. *IEEE Trans. Comput*. Vol. C-17, N 10, pp. 954-977, 1968