Web-Based System for Sequential Machines Decomposition

Sergei Devadze, Elena Fomina, Margus Kruus, and Alexander Sudnitson
Tallinn Technical University
Raja 15, 12618 Tallinn
Estonia
email: alsu@cc.ttu.ee

[1]*Abstract* - **This work focuses on particular but comprehensive problem of finite state machine (FSM) decomposition. The task of the FSM decomposition is essential to sequential circuits design optimization in implementation-independent manner. The main goal of the investigations has been to elaborate decomposition synthesis methods for high complexity FSMs and their implementation as Web-based computer design system. The theoretical basis for the investigations has been the algebraic structure theory of FSMs, its further development in accordance with the needs of digital systems design practice to handle the task of partition of hardware description into a network of interconnected FSMs targeting optimization criteria. Consideration of decomposition synthesis leads to investigation of hard NP-complete combinatorial problems. The synthesis system under development should not be only design automation software but it should be a research tool and educational system.**

Index Terms - **Decomposition, finite state machine, low-power design.**

I. Introduction

The task of decomposition has been a classic problem of discrete system theory for many years. The decomposition of FSM is a topic that waxes and wanes in importance. The fundamental works were done in the 1960s, became less interesting during the era beginning of era of VLSI, and is becoming more important again with pervasive use of programmable logic and low power applications in digital design. A large hardware behavioral description is decomposed into several smaller ones. One goal is to make the synthesis problem more tractable by providing smaller sub-problems that can be solved efficiently. Another goal is to create descriptions that can be synthesized into a structure that meets the design constraints. In the past, the synthesis focused on quality measures based on area and performance. The continuing decrease in feature size and increase in chip density in recent years have given rise to consider decomposition theory for low power as new dimension of the design process.

Various techniques have been developed to enhance the capability and efficiency of decomposition, and they fall broadly into two categories: those based on the algebraic theory [1] and those based on the factorization or on the identification in the state transition graph of subroutines [2].

During the last ten years, researches on decomposition techniques for low power have intensified; a range of techniques has been proposed for the optimization of circuits for low power using the second approach [3]-[5]. Theoretical background of our system is the algebraic structure theory of sequential machines, which uses partition pair algebra proposed in [1]. The importance of this theory lies in the fact that it provides a direct link between algebraic relationships and physical realizations of FSMs. The mathematical foundation of this theory rest on an algebraization of the concept of "information" in a machine and supply the algebraic formalism necessary to study problems about the flow of this information in machines as they operate. Unlike previous works [6], which focus on the modeling of informational flows of individual FSM, in this paper we consider a technique to evaluate the FSM decomposition with respect to the power consumption, which can be expected for the resulting network.

The problem of quality-driven synthesis corresponds to an optimal decomposition of a state machine reduced to choice of partitions on the set of states of prototype machine. Our work proceeds from the fact that the principal NP-complete problem of digital system decomposition is searching of partition set on the set of states of source specification that is substantial limitation of application of powerful algebraic decomposition theory in practice.

The paper is organized as follows. Section II describes the basic aspects of decomposition synthesis. In Section III, we formalize the information theoretic approach for quality relationship measures. The interactive system based on Java applets is described in Section IV. Finally, some experimental results and concluding remarks of the work discussed in Section V.

II. Preliminaries

Formally, the FSM is defined as a quintuple
$< S, X, Y, \delta, \lambda >$ where
$S = \{ s_1, \dots , s_M \}$ is a set of states;
$X = \{x_1, \dots , x_L \}$ is a set of binary input variables;

This work has been supported by the Estonian Science Foundation (under Grant 5643) and by EU V Framework projects IST-2001-37592 and IST-2000-30193.

$Y = \{y_1, \ldots, y_T\}$ is a set of binary output variables;
δ: $D(\delta) \rightarrow S$ is a multiple valued next state function with domain $D(\delta) = D_1 \times \ldots \times D_L \times S$ and codomain S,
$D_I = \{0,1\}$ represents a set of values each input variable x_i may assume;
λ: $D(\lambda) \rightarrow R(\lambda)$ is an output function with domain $D(\lambda) = D(\delta)$ and codomain $R(\lambda) = E_1 \times \ldots \times E_T$.
$E_i = \{0, 1\}$ represents a set of values each output variable y_i may assume.

The behavior of a control sequential component can be described by STG or, equivalently, by presentation by the list of transitions.

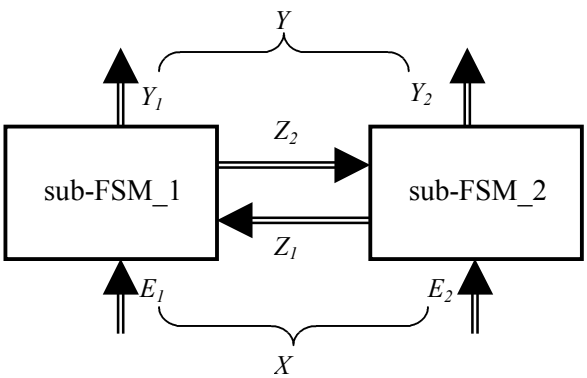

Fig.1. A decomposition of a FSM into two sub-FSMs

Let's we have an FSM which need to decompose in a network. Informally, the essence of the decomposition task could be described as follows. Given a prototype FSM description of a desired terminal behavior, the decomposition problem is to find sub-FSMs which, when interconnected in a prescribed way, will display that terminal behavior. Our procedure of the decomposition is based on the general form of the decomposition without the restriction on their interconnection (Fig. 1). Each sub-machine corresponds to a partition on the set of states (a partition π on the set of states, S, in a machine is a collection of disjoint subsets of states whose set union is S).

The state behavior of the FSMs network [1] forms the basis of decomposition model. The state behavior of prototype machine formally is described by the network of state machines $A_i = < X_i, S_i, \delta_i >$ where S_i is the set states elements of which correspond to blocks of partition π_i.
$X_i = Z_i \cup E_i$, where Z_i is a set of internal symbolic variables (state variables) and $E_i \subseteq X$ is a set of external inputs. Each of the sub-machines receives, as inputs, not only the primary inputs and its own state variables, but also the state variables of the other sub-machine. To describe the network more thoroughly, we use the set of internal symbolic variables of net $Z = \{z_i \mid z_i \in S_i, i \in I = \{1,\ldots,n\}\}$.
δ_i: $D(\delta_i) \rightarrow S_i$, is a transition function.

III. Quality relationship measures

A decomposition technique must define the attributes of a partition that determine the partition's "goodness". Such attributes are called quality relationship measures. Estimation algorithms must be efficient to be useful the design space exploration. If the complexity of an estimation algorithm approaches the complexities of the synthesis algorithms, then estimation is useless, since real designs can be generated as quickly as estimates. It is time consuming to generate completely synthesized designs for comparison of their quality. Hence, more important than the final quality measures are estimates of those measures early during the process when the final design is not available. Estimation attempts to predict the results of subsequent design steps.

In our reasoning, we proceed from information theoretic concepts [6], which are rationalized on the basis of algebraic structure theory of sequential machine. Interaction of several machines can be imagined by mutual information dependence that is reflected by elements of pair algebra. It is well known that entropy is a measure of information content. We have been trying to apply Shannon's entropy to estimate information flows in FSMs. By adapting standard notion of entropy, we interpret the partition entropy as a quantitative evaluation of information contents in this partition. On the one hand it is the next try to construct quantitative modeling of information in FSMs networks, based on the entropic relationships. Moreover, it is the application of the FSM decomposition to the problem of power consumption.

In the following, we assume that the state lines of the FSM are modeled as Markov chain characterized by the stochastic matrix $(q_{ij})_{1 \le i, j \le m}$, where q_{ij} is the conditional probability of the FSM being in j-th state given that it was previously in i-th state. These probabilities, along with the steady state probability vector $(p_i)_{1 \le i \le m}$ (we suppose that all states are reachable) can be found using standard techniques for probabilistic analysis of FSMs [7].

Let $E = \{e_1, e_2, \ldots e_g\}$ be a complete set of events which may occur with the probabilities $p_1, p_2, \ldots, p_g$ In order to quantify the content of information Shannon introduces the concept of entropy.

Entropy of E (denoted $H(E)$) is given by

$$H(E) = -\sum_{e \in E} p(e) \cdot \log_2 p(e) \tag{1}$$

Depending on the specified sense of event, we can define several entropy measures, e.g. the entropy of an FSM based on the state of occupation probabilities or based on the state transition probabilities. Reasoning similarly, we define the entropy of partition π as

$$H(\pi) = -\sum_{B \in \pi} p(B) \cdot \log_2 p(B) \tag{2}$$

where the probability of the block $B \subseteq S$ is defined as the cumulative occupation probability of the states in B.

Entropy of FSMs network corresponding to the set of partitions, N, *is* equal to

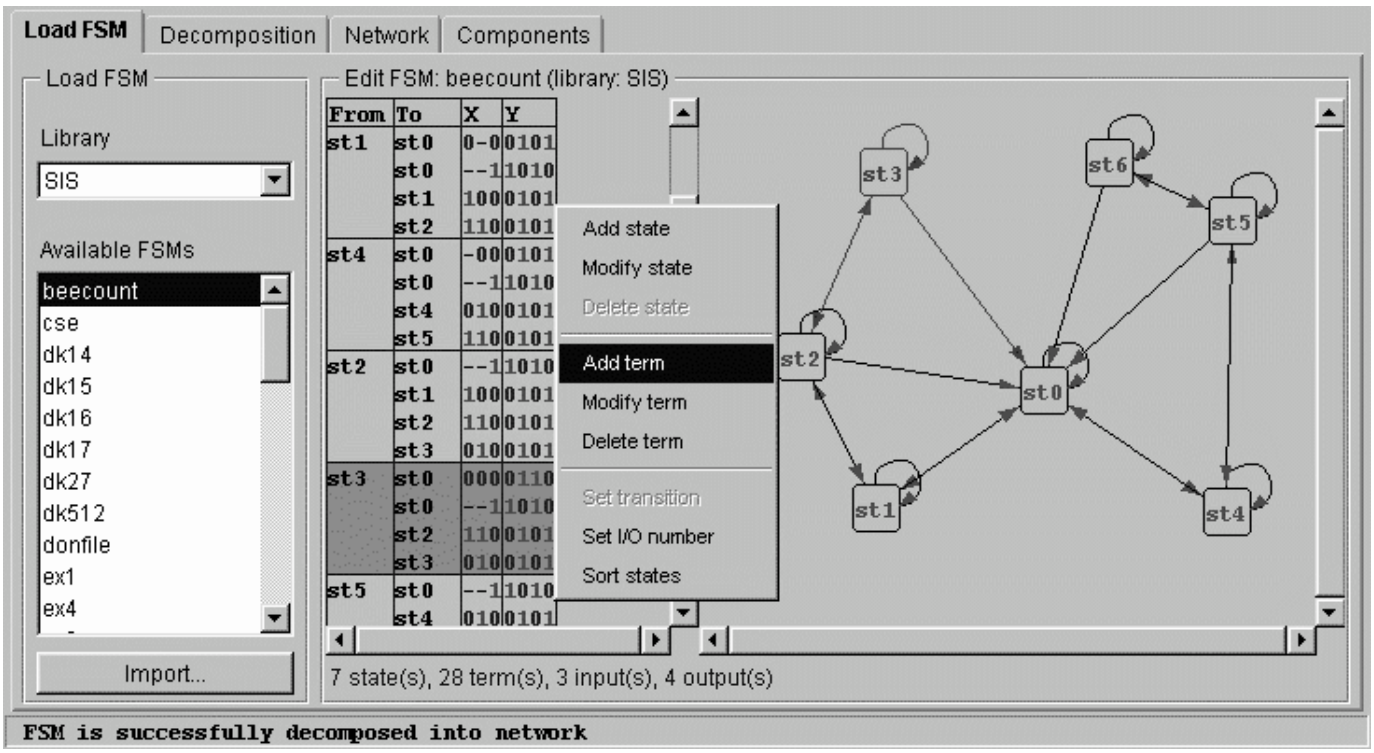

Fig. 2 An example applet

$$H(N) = \sum_{\pi \in N} H(\pi) \tag{3}$$

Entropy is related to switching activity, which is if the signal switching is high, it is likely that entropy is high also [8]. Theoretically confirmed high correlation proves that the partition entropy is suitable for estimating corresponding sub-machines, which makes it a good measure for partition choice for low power decomposition synthesis. For the estimation of switching activity of FSM as complete set of events, we consider the set of all transitions (corresponding to edges of STG) in this FSM. Proposed measures enable analysis of the information structure and information flows of a FSMs network to control low-power synthesis of a sequential circuit

IV. Software system

To implement the software system's architecture we should follow four main requirements [9] :

- Possibility to ran under various operating systems
- Implementation of new modules without changing the rest of the system
- Realizing a client server architecture
- Using the same source to generate the printed and interactive worksheets to prevent inconsistency after modifications.

These requirements cause the use the applet concept of Java language. Java is the natural programming language of choice on the client side because of its flexibility of GUI design, convenient network programming, and platform independence. The last property is especially significant since it allows the same applet program to run on client computers of different platform.

Developed system includes building tutorial of the FSM decomposition theory and additional useful information for working with client software. The advantage of the tutorial is interconnectedness among different topics and with related tutorials, which is easy to implement on the WWW using the hypertext mark-up language.

The sequence of applets is developed. Next, we discuss main of them from "informational" point of view. Let us break them down into three groups.

Group 1 helps us to understand the essence of the FSM decomposition problem. The first applet describes how partitions on a set can be "multiplied" and "added". These operations on partitions play a central role in the structure theory of FSM and form a basic link between machine concepts and algebra. The sum of two partitions π_1 and π_2 is the smallest partition that is refined by both π_1 and π_2. The product of π_1 and π_2 is the largest partition that refines both π_1 and π_2. A partition on the set of states of the FSM can be considered as a measure of information about the FSM. That it is why the multiplication of all partitions in the set must be zero partition in order to preserve all the information about the source FSM behavior in the network of FSMs defined by the partition set. The functionality of the prototype machine is maintained in the decomposed machine if the partitions associated with the decomposition are such that their product is the zero-partition on S (every block of partition consists exactly of one state). The next applet exhibits formal

correspondence to intuitive concept of a ”subcomputation”. We consider the concept of a homomorphism. Since a machine A can be used to realize its homomorphic image A', we can say informally that A' does a part or a subcomputation of the computation performed by A. From partition algebra point of view the concept of homomorphism relates to partitions with substitution property. We recall that if a partition π on the set of states of a machine A has the substitution property, than as long as we know the block of π which contains a given state of A, we can compute the block of π to which that state is transformed by any given input sequence. Intuitively we say that the “ignorance” about the given state (as specified by the partition π) does not spread as the machine operates [1]. The concept of partition pairs is more general than substitution property and is introduced to study how “ignorance spreads” or “information flows” through a sequential machine when it operates. If (π,π') is a partition pair on the FSM A than blocks of π are mapped into the blocks of π' by A In other words, if we only know the block of π which contains the state of A, then we can compute for every input the block of π' to which this state is transferred by A.

For partition pair $\langle \pi_i, \pi_j \rangle$ the conditional entropy is

$$H(\pi_j,\pi_i) = H(\pi_i \cdot \pi_j) - H(\pi_i) \tag{4}$$

The concept of partition pair and its “informational representation” is presented by corresponding applet.

It is natural that for any partition π we can determine the $M(\pi)$ partition. The operator $M(\pi)$ gives the maximum front partition of partition pair. Informally speaking, for a given partition π, the partition $M(\pi)$ describes the least amount of information we must have about the present state of A to the next state (i.e., the block of π which contains the next state of A). Thus these partitions give precise meaning to our intuitive concept “how much do we have to know about the present state to compute … about the next state”.

To calculate this partition we need to find the symbolic cover of the discrete function F_i: $D(\delta) \rightarrow \pi_i$. Given a FSM, we first assign one-hot codes to all states. Then symbolic minimization is applied to the one-hot coded machine using multi-valued logic minimization. The result is a symbolic cover, K_i, of the F_i. Each element of the symbolic cover is a symbolic prime implicant, that is a triple $\langle \beta, B', B \rangle$ where B' is the set of states (block of partition $M(\pi)$) which transit to the next state contained in the same block B of partition π under input condition β. The number of prime implicates, $|K_i|$, is proportional to number of rows in the transition table of corresponding sub-machine.

Our work proceeds from the fact that the principal NP-hard problem of FSM decomposition is searching of a set of partitions on the set of states of prototype FSM. As it was shown in [1], only such a set of partitions may be used for the FSM decomposition. Implementation of FSM in a device with the lack of external terminals appears very often in practice and has always been a problem for designers.

Group 2 of applets performs construction of FSMs network that realizes the prototype FSM. We represent a relation of connection of sub-FSM in the network as incidence matrix $\|r_{ij}\|$. $r_{ij} = 1$ means i-th component FSM receives information from j-th component FSM. Every FSM from B is in correspondence with chosen partition π_i. If partition $\tau=M(\pi_i)$ is less or equal to multiplication of partitions from $\{\pi_j \mid r_{ij} = 1\}$ than it means that i-th component FSM receives enough information from component FSMs with which it is connected accordingly F to compute the next state.

The construction of FSMs network is based on decomposition partitions on the set of states of prototype FSM. The group of applets is devoted to choice of decomposition partitions to meet a requirement on the distribution of primary inputs and outputs among sub-FSMs. Here we should emphasize the fact that the machine decomposition problem and the reduction of variable dependence are virtually identical concepts. This is NP-hard problem, and amounts to solving a face hypercube-embedding problem [2, 10]. In spite of recent advances, computing a decision of this task remains prohibitive for FSM of practical complexity. In this applet we show how the input-state dependencies can be used to decrease the number of inputs. Theoretical foundation of our approach is based on the new notion of partition with don’t care’s and its relation to pair algebra.

The idea of applets of *Group 3* is to introduce additional “idle” states into the FSM in the hope to meet power design constraints. The network of FSMs consists of components working alternatively in time, i.e. all components except one are suspended in one of extra state (the “wait” state). In [2], similar approach is called factorization of the sequential state machines. This property gives opportunity to apply sleep mode operation (dynamic power management) for saving power consumption. Corresponding applet enables to decompose a prototype FSM into a set connected component FSMs with given constraints on the complexity of component FSMs (a number of inputs, outputs, states and rows in their transition tables) on the base of one partition on the set of states. The number of states of component FSM is equal to the number of states in corresponding block of partition π plus 1 (wait or idle state).

V. Concluding Remarks

The developed Web-based design system can be considered as a research tool that we use to carry out experiments guided to further development of the decomposition synthesis. Experiments have been carried out on the range of random FSMs and on the set of well-known FSM benchmarks [11] to certify the viability of our concepts. For example, some results of experiments are presented in Table I. This table contains the results of comparative experiments of our decomposition technique and approach used in [3]-[5]. The area estimation was done using the

commercial design frame (SYNOPSIS). This parameter was chosen for complexity criteria for decomposition system.

TABLE I
RESULTS AND COMPARISON OF IMPLEMENTATIONS

Prototype machine	Total # of states of sub-FSMs	Total # of states (alt. approach)	Area combinat. ratio
log	12	19	0.75
dvram	14	37	0.50
nucpw	15	31	0.68
sync	26	54	0.67
planet	24	50	0.59
ex6	9	10	1.14
opus	7	12	0.78
ex4	9	16	0.75

The promising application of our technique is low power design of control-dominated discrete systems. The idea of partition for low power is that in behavioral descriptions of hardware, a small set of computation often accounts for most of the computational complexity as well as power dissipation. The decomposition focuses on power dissipation as the main criteria of design optimization. Techniques based on disabling the input/state registers when some input conditions are met have been proposed and shown to be among the most effective in reducing the overall switching activity in sequential circuits.

Our reasoning proceeds from the premise that the solution of the problem of FSM synthesis for low power can be reduced to the FSM decomposition with distributed primary output/input variables and appropriate synthesis of FSMs network. Results confirmed that it is possible to significantly reduce switching activity of implementation and that significant reduction in power consumption could be achieved without performance degradation.

We have been examining the entropy measures for search an approximate or indirect methods of evaluation of power consumption. What makes entropy especially useful from decomposition point of view, is fact, that it provides a direct link between quality relationships and physical realizations of machines. The idea of using entropy based informational measures can be extended to other phases of logic synthesis also. To ensure that the partition entropy is a good indicator of implementation complexity, experiments have been carried on hundreds of FSMs. They proved that the correlation between of the decomposition partition and the complexity (area) of corresponding sub-FSM is very high.

It is expected to incorporate developed investigations and design system in educational process for interactive remote distance learning.

REFERENCES

[1] Hartmanis J. and R. E. Stearns, *Algebraic Structure Theory of Sequential Machines*, Englewood Cliffs, N.J.: Prentice-Hall, 1966.

[2] P. Ashar, S.Devadas, and A.R. Newton, *Sequential Logic Synthesis*, Kluwer Academic Publishers, Boston, 1992.

[3] J.C. Monteiro and A.L.Oliveira, "Implicit FSM Decomposition Applied to Low-Power Design," *IEEE Trans. on VLSI Systems*, vol.10, No. 5, pp.560-565, October 2002.

[4] E.Hwang, F.Vahid, and Y.-C.Hsu, "FSMD Functional Partitioning for Low Power", in *Proc. DATE Conf.*, pp.22-28, March 1999.

[5] M.H. Lee, T.T. Hwang, and S.-Y. Huang, "Decomposition of Extended Finite State Machine for Low Power Design," in *Proc. DATE Conf.*, 2003, pp. 1152 – 1154.

[6] Jozwiak, L., Slusarczyk, A., and A. Choinacki,. "Fast and Compact Sequential Circuits Through the Information-Driven Synthesis,". In: *Proceedings of the 27th Euromicro Conference*, Poland, pp.46-53, 2001.

[7] Hachtel G. D., Macii E., Pardo A., and F. Somenzi, "Markovian Analysis of Large Finite State Machine," *IEEE Trans. Computer-Aided Design*, Vol. 15, pp. 1479-1493, 1996.

[8] Marculescu D., Marculescu R., and M. Pedram, "Information Theoretic Measures for Power Analysis," *IEEE Trans. Computer-Aided Design*, Vol. 15, 1996, pp. 599-610.

[9] Wuttke H.-D., Henke K., R. Peukert, "Internet Based Education : An Experimental Environment for Various Educational Purposes," *Proc. of the IASTED Int. Conf. on Computers and Advanced Technology in Education*, Philadelphia, PA USA. IASTED/Acta Press No. 292, 1999, pp. 50-54.

[10] A. Sudnitson, "Computational Kernel Extraction for Synthesis of Power-Managed Sequential Components," in *Proc. IEEE 9th International Conference on Electronics, Circuits and Systems*, Dubrovnik, Croatia, pp. 749-752, 2002.

[11] McElvain, K.: LGSynth'93 Benchmark Set: Version 4.0, 1993. Available: http://www.cbl.ncsu.edu/benchmarks.

[12] Decomposition Applets Available: http://www.pld.ttu.ee/dildis/automata/applets

Web-Based System for Sequential Machines Decomposition

Sergei Devadze, Elena Fomina, Margus Kruus, and Alexander Sudnitson
Tallinn Technical University
Raja 15, 12618 Tallinn
Estonia
email: alsu@cc.ttu.ee

[1]*Abstract* - **This work focuses on particular but comprehensive problem of finite state machine (FSM) decomposition. The task of the FSM decomposition is essential to sequential circuits design optimization in implementation-independent manner. The main goal of the investigations has been to elaborate decomposition synthesis methods for high complexity FSMs and their implementation as Web-based computer design system. The theoretical basis for the investigations has been the algebraic structure theory of FSMs, its further development in accordance with the needs of digital systems design practice to handle the task of partition of hardware description into a network of interconnected FSMs targeting optimization criteria. Consideration of decomposition synthesis leads to investigation of hard NP-complete combinatorial problems. The synthesis system under development should not be only design automation software but it should be a research tool and educational system.**

Index Terms - **Decomposition, finite state machine, low-power design.**

I. Introduction

The task of decomposition has been a classic problem of discrete system theory for many years. The decomposition of FSM is a topic that waxes and wanes in importance. The fundamental works were done in the 1960s, became less interesting during the era beginning of era of VLSI, and is becoming more important again with pervasive use of programmable logic and low power applications in digital design. A large hardware behavioral description is decomposed into several smaller ones. One goal is to make the synthesis problem more tractable by providing smaller sub-problems that can be solved efficiently. Another goal is to create descriptions that can be synthesized into a structure that meets the design constraints. In the past, the synthesis focused on quality measures based on area and performance. The continuing decrease in feature size and increase in chip density in recent years have given rise to consider decomposition theory for low power as new dimension of the design process.

Various techniques have been developed to enhance the capability and efficiency of decomposition, and they fall broadly into two categories: those based on the algebraic theory [1] and those based on the factorization or on the identification in the state transition graph of subroutines [2].

During the last ten years, researches on decomposition techniques for low power have intensified; a range of techniques has been proposed for the optimization of circuits for low power using the second approach [3]-[5]. Theoretical background of our system is the algebraic structure theory of sequential machines, which uses partition pair algebra proposed in [1]. The importance of this theory lies in the fact that it provides a direct link between algebraic relationships and physical realizations of FSMs. The mathematical foundation of this theory rest on an algebraization of the concept of "information" in a machine and supply the algebraic formalism necessary to study problems about the flow of this information in machines as they operate. Unlike previous works [6], which focus on the modeling of informational flows of individual FSM, in this paper we consider a technique to evaluate the FSM decomposition with respect to the power consumption, which can be expected for the resulting network.

The problem of quality-driven synthesis corresponds to an optimal decomposition of a state machine reduced to choice of partitions on the set of states of prototype machine. Our work proceeds from the fact that the principal NP-complete problem of digital system decomposition is searching of partition set on the set of states of source specification that is substantial limitation of application of powerful algebraic decomposition theory in practice.

The paper is organized as follows. Section II describes the basic aspects of decomposition synthesis. In Section III, we formalize the information theoretic approach for quality relationship measures. The interactive system based on Java applets is described in Section IV. Finally, some experimental results and concluding remarks of the work discussed in Section V.

II. Preliminaries

Formally, the FSM is defined as a quintuple
$< S, X, Y, \delta, \lambda >$ where
$S = \{ s_1, \dots , s_M \}$ is a set of states;
$X = \{x_1, \dots , x_L \}$ is a set of binary input variables;

This work has been supported by the Estonian Science Foundation (under Grant 5643) and by EU V Framework projects IST-2001-37592 and IST-2000-30193.

$Y = \{y_1, \ldots, y_T\}$ is a set of binary output variables;
δ: $D(\delta) \rightarrow S$ is a multiple valued next state function with domain $D(\delta) = D_1 \times \ldots \times D_L \times S$ and codomain S, $D_I = \{0,1\}$ represents a set of values each input variable x_i may assume;
λ: $D(\lambda) \rightarrow R(\lambda)$ is an output function with domain $D(\lambda) = D(\delta)$ and codomain $R(\lambda) = E_1 \times \ldots \times E_T$.
$E_i = \{0, 1\}$ represents a set of values each output variable y_i may assume.

The behavior of a control sequential component can be described by STG or, equivalently, by presentation by the list of transitions.

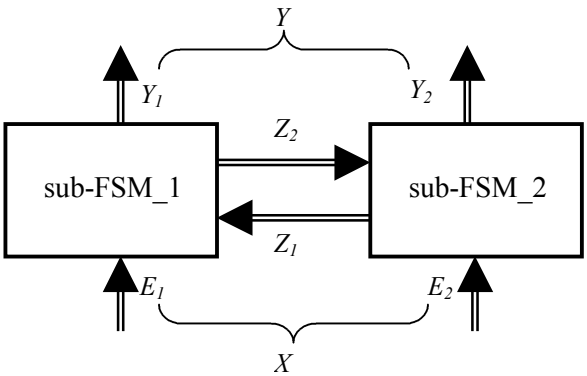

Fig.1. A decomposition of a FSM into two sub-FSMs

Let's we have an FSM which need to decompose in a network. Informally, the essence of the decomposition task could be described as follows. Given a prototype FSM description of a desired terminal behavior, the decomposition problem is to find sub-FSMs which, when interconnected in a prescribed way, will display that terminal behavior. Our procedure of the decomposition is based on the general form of the decomposition without the restriction on their interconnection (Fig. 1). Each sub-machine corresponds to a partition on the set of states (a partition π on the set of states, S, in a machine is a collection of disjoint subsets of states whose set union is S).

The state behavior of the FSMs network [1] forms the basis of decomposition model. The state behavior of prototype machine formally is described by the network of state machines $A_i = < X_i, S_i, \delta_i >$ where S_i is the set states elements of which correspond to blocks of partition π_i.
$X_i = Z_i \cup E_i$, where Z_i is a set of internal symbolic variables (state variables) and $E_i \subseteq X$ is a set of external inputs. Each of the sub-machines receives, as inputs, not only the primary inputs and its own state variables, but also the state variables of the other sub-machine. To describe the network more thoroughly, we use the set of internal symbolic variables of net $Z = \{z_i \mid z_i \in S_i, i \in I = \{1,\ldots,n\}\}$.
δ_i: $D(\delta_i) \rightarrow S_i$, is a transition function.

III. Quality relationship measures

A decomposition technique must define the attributes of a partition that determine the partition's "goodness". Such attributes are called quality relationship measures. Estimation algorithms must be efficient to be useful the design space exploration. If the complexity of an estimation algorithm approaches the complexities of the synthesis algorithms, then estimation is useless, since real designs can be generated as quickly as estimates. It is time consuming to generate completely synthesized designs for comparison of their quality. Hence, more important than the final quality measures are estimates of those measures early during the process when the final design is not available. Estimation attempts to predict the results of subsequent design steps.

In our reasoning, we proceed from information theoretic concepts [6], which are rationalized on the basis of algebraic structure theory of sequential machine. Interaction of several machines can be imagined by mutual information dependence that is reflected by elements of pair algebra. It is well known that entropy is a measure of information content. We have been trying to apply Shannon's entropy to estimate information flows in FSMs. By adapting standard notion of entropy, we interpret the partition entropy as a quantitative evaluation of information contents in this partition. On the one hand it is the next try to construct quantitative modeling of information in FSMs networks, based on the entropic relationships. Moreover, it is the application of the FSM decomposition to the problem of power consumption.

In the following, we assume that the state lines of the FSM are modeled as Markov chain characterized by the stochastic matrix $(q_{ij})_{1 \le i, j \le m}$, where q_{ij} is the conditional probability of the FSM being in j-th state given that it was previously in i-th state. These probabilities, along with the steady state probability vector $(p_i)_{1 \le i \le m}$ (we suppose that all states are reachable) can be found using standard techniques for probabilistic analysis of FSMs [7].

Let $E = \{e_1, e_2, \ldots e_g\}$ be a complete set of events which may occur with the probabilities $p_1, p_2, \ldots, p_g$ In order to quantify the content of information Shannon introduces the concept of entropy. Entropy of E (denoted $H(E)$) is given by

$$H(E) = -\sum_{e \in E} p(e) \cdot \log_2 p(e) \tag{1}$$

Depending on the specified sense of event, we can define several entropy measures, e.g. the entropy of an FSM based on the state of occupation probabilities or based on the state transition probabilities. Reasoning similarly, we define the entropy of partition π as

$$H(\pi) = -\sum_{B \in \pi} p(B) \cdot \log_2 p(B) \tag{2}$$

where the probability of the block $B \subseteq S$ is defined as the cumulative occupation probability of the states in B.

Entropy of FSMs network corresponding to the set of partitions, N, *is* equal to

$$H(N) = \sum_{\pi \in N} H(\pi) \tag{3}$$

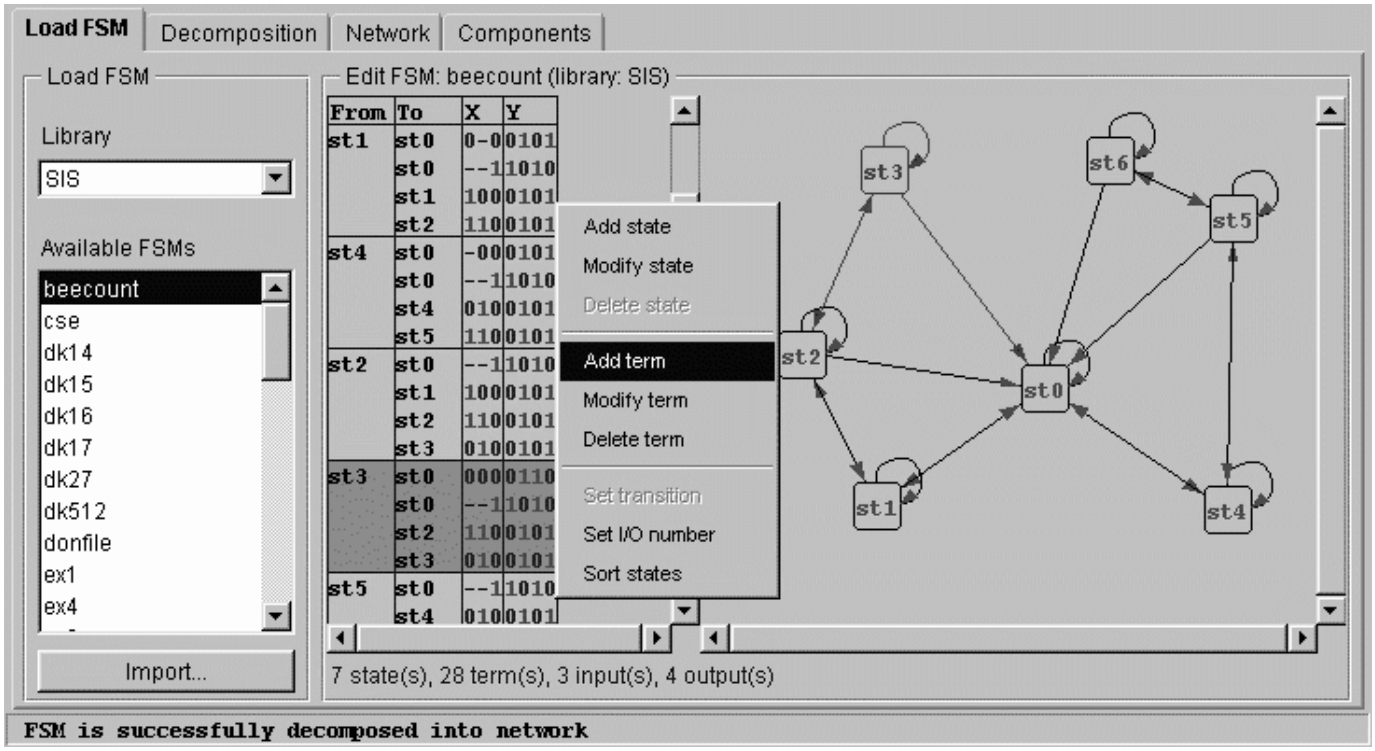

Fig. 2 An example applet

Entropy is related to switching activity, which is if the signal switching is high, it is likely that entropy is high also [8]. Theoretically confirmed high correlation proves that the partition entropy is suitable for estimating corresponding sub-machines, which makes it a good measure for partition choice for low power decomposition synthesis. For the estimation of switching activity of FSM as complete set of events, we consider the set of all transitions (corresponding to edges of STG) in this FSM. Proposed measures enable analysis of the information structure and information flows of a FSMs network to control low-power synthesis of a sequential circuit

IV. Software system

To implement the software system's architecture we should follow four main requirements [9] :

- Possibility to ran under various operating systems
- Implementation of new modules without changing the rest of the system
- Realizing a client server architecture
- Using the same source to generate the printed and interactive worksheets to prevent inconsistency after modifications.

These requirements cause the use the applet concept of Java language. Java is the natural programming language of choice on the client side because of its flexibility of GUI design, convenient network programming, and platform independence. The last property is especially significant since it allows the same applet program to run on client computers of different platform.

Developed system includes building tutorial of the FSM decomposition theory and additional useful information for working with client software. The advantage of the tutorial is interconnectedness among different topics and with related tutorials, which is easy to implement on the WWW using the hypertext mark-up language.

The sequence of applets is developed. Next, we discuss main of them from "informational" point of view. Let us break them down into three groups.

Group 1 helps us to understand the essence of the FSM decomposition problem. The first applet describes how partitions on a set can be "multiplied" and "added". These operations on partitions play a central role in the structure theory of FSM and form a basic link between machine concepts and algebra. The sum of two partitions π_1 and π_2 is the smallest partition that is refined by both π_1 and π_2. The product of π_1 and π_2 is the largest partition that refines both π_1 and π_2. A partition on the set of states of the FSM can be considered as a measure of information about the FSM. That it is why the multiplication of all partitions in the set must be zero partition in order to preserve all the information about the source FSM behavior in the network of FSMs defined by the partition set. The functionality of the prototype machine is maintained in the decomposed machine if the partitions associated with the decomposition are such that their product is the zero-partition on S (every block of partition consists exactly of one state). The next applet exhibits formal correspondence to intuitive concept of a "subcomputation". We consider the concept of a homomorphism. Since a machine A can be used to realize its homomorphic image A',

we can say informally that *A'* does a part or a subcomputation of the computation performed by *A*. From partition algebra point of view the concept of homomorphism relates to partitions with substitution property. We recall that if a partition π on the set of states of a machine *A* has the substitution property, than as long as we know the block of π which contains a given state of *A*, we can compute the block of π to which that state is transformed by any given input sequence. Intuitively we say that the "ignorance" about the given state (as specified by the partition π) does not spread as the machine operates [1]. The concept of partition pairs is more general than substitution property and is introduced to study how "ignorance spreads" or "information flows" through a sequential machine when it operates. If (π,π') is a partition pair on the FSM *A* than blocks of π are mapped into the blocks of π' by *A* In other words, if we only know the block of π which contains the state of *A*, then we can compute for every input the block of π' to which this state is transferred by *A*.

For partition pair $\langle \pi_i, \pi_j \rangle$ the conditional entropy is

$$H(\pi_j,\pi_i) = H(\pi_i \cdot \pi_j) - H(\pi_i) \tag{4}$$

The concept of partition pair and its "informational representation" is presented by corresponding applet.

It is natural that for any partition π we can determine the $M(\pi)$ partition. The operator $M(\pi)$ gives the maximum front partition of partition pair. Informally speaking, for a given partition π, the partition $M(\pi)$ describes the least amount of information we must have about the present state of *A* to the next state (i.e., the block of π which contains the next state of *A*). Thus these partitions give precise meaning to our intuitive concept "how much do we have to know about the present state to compute … about the next state".

To calculate this partition we need to find the symbolic cover of the discrete function F_i: $D(\delta) \rightarrow \pi_i$. Given a FSM, we first assign one-hot codes to all states. Then symbolic minimization is applied to the one-hot coded machine using multi-valued logic minimization. The result is a symbolic cover, K_i, of the F_i. Each element of the symbolic cover is a symbolic prime implicant, that is a triple $\langle \beta, B', B \rangle$ where *B'* is the set of states (block of partition $M(\pi)$) which transit to the next state contained in the same block *B* of partition π under input condition β. The number of prime implicates, $|K_i|$, is proportional to number of rows in the transition table of corresponding sub-machine.

Our work proceeds from the fact that the principal NP-hard problem of FSM decomposition is searching of a set of partitions on the set of states of prototype FSM. As it was shown in [1], only such a set of partitions may be used for the FSM decomposition. Implementation of FSM in a device with the lack of external terminals appears very often in practice and has always been a problem for designers.

Group 2 of applets performs construction of FSMs network that realizes the prototype FSM. We represent a relation of connection of sub-FSM in the network as incidence matrix $\|r_{ij}\|$. $r_{ij} = 1$ means *i*-th component FSM receives information from *j*-th component FSM. Every FSM from *B* is in correspondence with chosen partition π_i. If partition $\tau=M(\pi_i)$ is less or equal to multiplication of partitions from $\{\pi_j \mid r_{ij} = 1\}$ than it means that *i*-th component FSM receives enough information from component FSMs with which it is connected accordingly *F* to compute the next state.

The construction of FSMs network is based on decomposition partitions on the set of states of prototype FSM. The group of applets is devoted to choice of decomposition partitions to meet a requirement on the distribution of primary inputs and outputs among sub-FSMs. Here we should emphasize the fact that the machine decomposition problem and the reduction of variable dependence are virtually identical concepts. This is NP-hard problem, and amounts to solving a face hypercube-embedding problem [2, 10]. In spite of recent advances, computing a decision of this task remains prohibitive for FSM of practical complexity. In this applet we show how the input-state dependencies can be used to decrease the number of inputs. Theoretical foundation of our approach is based on the new notion of partition with don't care's and its relation to pair algebra.

The idea of applets of *Group 3* is to introduce additional "idle" states into the FSM in the hope to meet power design constraints. The network of FSMs consists of components working alternatively in time, i.e. all components except one are suspended in one of extra state (the "wait" state). In [2], similar approach is called factorization of the sequential state machines. This property gives opportunity to apply sleep mode operation (dynamic power management) for saving power consumption. Corresponding applet enables to decompose a prototype FSM into a set connected component FSMs with given constraints on the complexity of component FSMs (a number of inputs, outputs, states and rows in their transition tables) on the base of one partition on the set of states. The number of states of component FSM is equal to the number of states in corresponding block of partition π plus 1 (wait or idle state).

V. CONCLUDING REMARKS

The developed Web-based design system can be considered as a research tool that we use to carry out experiments guided to further development of the decomposition synthesis. Experiments have been carried out on the range of random FSMs and on the set of well-known FSM benchmarks [11] to certify the viability of our concepts. For example, some results of experiments are presented in Table I. This table contains the results of comparative experiments of our decomposition technique and approach used in [3]-[5]. The area estimation was done using the commercial design frame (SYNOPSIS). This parameter was chosen for complexity criteria for decomposition system.

TABLE I
RESULTS AND COMPARISON OF IMPLEMENTATIONS

Prototype machine	Total # of states of sub-FSMs	Total # of states (alt. approach)	Area combinat. ratio
log	12	19	0.75
dvram	14	37	0.50
nucpw	15	31	0.68
sync	26	54	0.67
planet	24	50	0.59
ex6	9	10	0.99
opus	7	12	0.78
ex4	9	16	0.75

The promising application of our technique is low power design of control-dominated discrete systems. The idea of partition for low power is that in behavioral descriptions of hardware, a small set of computation often accounts for most of the computational complexity as well as power dissipation. The decomposition focuses on power dissipation as the main criteria of design optimization. Techniques based on disabling the input/state registers when some input conditions are met have been proposed and shown to be among the most effective in reducing the overall switching activity in sequential circuits.

Our reasoning proceeds from the premise that the solution of the problem of FSM synthesis for low power can be reduced to the FSM decomposition with distributed primary output/input variables and appropriate synthesis of FSMs network. Results confirmed that it is possible to significantly reduce switching activity of implementation and that significant reduction in power consumption could be achieved without performance degradation.

We have been examining the entropy measures for search an approximate or indirect methods of evaluation of power consumption. What makes entropy especially useful from decomposition point of view, is fact, that it provides a direct link between quality relationships and physical realizations of machines. The idea of using entropy based informational measures can be extended to other phases of logic synthesis also. To ensure that the partition entropy is a good indicator of implementation complexity, experiments have been carried on hundreds of FSMs. They proved that the correlation between of the decomposition partition and the complexity (area) of corresponding sub-FSM is very high.

It is expected to incorporate developed investigations and design system in educational process for interactive remote distance learning.

REFERENCES

[1] Hartmanis J. and R. E. Stearns, *Algebraic Structure Theory of Sequential Machines*, Englewood Cliffs, N.J.: Prentice-Hall, 1966.

[2] P. Ashar, S.Devadas, and A.R. Newton, *Sequential Logic Synthesis*, Kluwer Academic Publishers, Boston, 1992.

[3] J.C. Monteiro and A.L.Oliveira, "Implicit FSM Decomposition Applied to Low-Power Design," *IEEE Trans. on VLSI Systems*, vol.10, No. 5, pp.560-565, October 2002.

[4] E.Hwang, F.Vahid, and Y.-C.Hsu, "FSMD Functional Partitioning for Low Power", in *Proc. DATE Conf.*, pp.22-28, March 1999.

[5] M.H. Lee, T.T. Hwang, and S.-Y. Huang, "Decomposition of Extended Finite State Machine for Low Power Design," in *Proc. DATE Conf.*, 2003, pp. 1152 – 1154.

[6] Jozwiak, L., Slusarczyk, A., and A. Choinacki,. "Fast and Compact Sequential Circuits Through the Information-Driven Synthesis,". In: *Proceedings of the 27th Euromicro Conference*, Poland, pp.46-53, 2001.

[7] Hachtel G. D., Macii E., Pardo A., and F. Somenzi, "Markovian Analysis of Large Finite State Machine," *IEEE Trans. Computer-Aided Design*, Vol. 15, pp. 1479-1493, 1996.

[8] Marculescu D., Marculescu R., and M. Pedram, "Information Theoretic Measures for Power Analysis," *IEEE Trans. Computer-Aided Design*, Vol. 15, 1996, pp. 599-610.

[9] Wuttke H.-D., Henke K., R. Peukert, "Internet Based Education : An Experimental Environment for Various Educational Purposes," *Proc. of the IASTED Int. Conf. on Computers and Advanced Technology in Education*, Philadelphia, PA USA. IASTED/Acta Press No. 292, 1999, pp. 50-54.

[10] A. Sudnitson, "Computational Kernel Extraction for Synthesis of Power-Managed Sequential Components," in *Proc. IEEE 9th International Conference on Electronics, Circuits and Systems*, Dubrovnik, Croatia, pp. 749-752, 2002.

[11] McElvain, K.: LGSynth'93 Benchmark Set: Version 4.0, 1993. Available: http://www.cbl.ncsu.edu/benchmarks.

[12] Decomposition Applets Available: http://www.pld.ttu.ee/dildis/automata/applets