Architecture Level Design Space Exploration and Mapping of Hardwares

Sujan Pandey, Manfred Glesner, Max Mühlhäuser[†]
Institute of Microelectronics Systems, Karlstr. 15, D-64283, Darmstadt
[†]Telecooperation Group, Hochschulstr. 10, D-64283, Darmstadt
Darmstadt University of Technology, Germany
{pandey,glesner}@mes.tu-darmstadt.de, max@informatik.tu-darmstadt.de

***Abstract*— After the partitioning in Hw/Sw co-design, an efficient mapping of hardware components to the target architectures fulfilling both power and delay requirements are still a challenging task for a system designer. In this paper, high level power/delay estimation to map hardware components to the target architectures based on petri net is proposed, which helps designers to model hardware architecture at high level and estimates power and delay for several design alternatives. This estimation shows how various solutions are distributed over the entire design space and helps to find an optimal solution. In this approach, we first extract parameters such as capacitances and resistances of a gate from the transistor level and form a library of basic components such as adder, multiplier, FIFO etc. of several sizes with their corresponding power/delay informations. We model a hardware architecture in petri net by taking necessary components from the library and estimate power/delay for several design alternatives. For an experimental purpose, we model a FFT hardware architecture and the results clearly demonstrate the utility of our techniques for the mapping of hardware based on power/delay informations.**

I. Introduction

The possibility of high level integration provided by the current technology has permitted the implementation of very complex systems on a single chip. To cope with design complexity, the design process begins at high level of abstraction, where no implementation details of the system are considered and a designer has possibilities of analyzing system characteristics.

At each level of abstraction in a system design flow, a designer has to face a large numbers of design choices such as low power, high performance, small in size, low cost, less time to market etc. If one of these choices is wrong that may lead to a bad design. To obtain the best solution considering several design alternatives, system characteristics must be analyzed at the higher level of abstraction [1,2,3,9]. At high level of abstraction, however absolute accuracy is impossible. Rather, relative accuracy (fidelity) [9] allows designers to prune design space of infeasible alternatives. Furthermore, in order to evaluate complete system architectures in a comprehensive and unified manner, it must be possible to estimate a wide variety of target implementations in combination. Also, a wide range of metrics for performance, traffic, storage, etc. should be available for use in different design domains.

In this paper, we focus our work to analyze system characteristics at the level just after the partitioning of system into hardware/software, assuming that the on chip communication structure has been already identified. At this level designer has a list of hardwares, which have to be mapped onto the target architectures. We propose a method of estimating power and delay of the target architecture for different design alternatives based on petri net. A brief introduction about petri net is presented in section 3 and details can be found in [6,7].

A hardware architecture is modeled in petri net by using the library of basic components such as adder, multiplier etc. This library is a set of components with estimated power and delay, which are represented in terms of probability density function. The estimation of power and delay are done by extracting parameters such as capacitances and resistances of gates from the transistor level. These values are estimated on the basis of signal probability of being logic ’1’ or ’0’ at the input of gate. The power of hardware component is obtained by the power equation $P = \frac{1}{2}V_{dd}^2 f C_{eff}$, where V_{dd} is the supply voltage, f is the operating frequency, C_{eff} is the effective load capacitance. Similarly, the delay of hardware is obtained by the delay equation $t_d = \frac{t_{PLH}+t_{PHL}}{2}$, where t_{PLH} is the charging time of effective load capacitance and t_{PHL} is the discharging time of the pull-down network. The details of these formulae are given in [10]. From the definition of petri net, each transition has one parameter e.g. delay but in our work to model both power and delay of a hardware, we modified the property of petri net by defining two parameters in each transition so that power/delay characteristics of a hardware can be assigned to each transition of the petri net.

II. Related work

There have been tremendous amount of works that are done in the area of design space exploration by estimating power, delay, size etc. of systems at both system and architecture level. In [1] system level power estimation and mapping techniques are proposed based on POLIS co-design environment, which is for a control dominated systems aiming at a processor-coprocessor architectures. They describe a system at the behavioral level as a set of concurrent communicating processes and this description is partitioned by POLIS, which generates netlist for hardware and codes for software to estimate the power. In [2], their method converts the target architecture in terms of gates equivalents and estimates only the power. In [8] they used Avalanche as a tool to estimate power by extracting the effective capacitances of hardware at the architecture level. In [3,5] they proposed power estimation techniques based on input signal switching activity.

Most of the above mapping approaches take into account only the power. They estimate parameter/s for a specific design, if that design does not satisfy the requirements then they change the design and estimate again until the requirement is not fulfilled, this is of course the most time taking task. In addition to this, above approaches also lack of estimating both power and delay for several design alternatives at a same time. In this paper, we propose a method to estimate power and delay for several design alternatives at a same time so that a designer can easily see different possible mapping of hardware. This avoids the designer to model and estimate system parameters for every possible designs from the scratch. We choose petri net for modeling hardware because they are hierarchical and intuitive to model a complex system. We model a hardware in the petri net by first extracting parameters such as capacitance and resistance of gates from the transistor level, considering additional signal switching probability at inputs. After this, we evaluate power and delay of each component like adder, multiplier, FIFO etc. of different size, which then act as a library for a target architecture. With these estimated basic components in the library, we model the target architecture in petri net, where power and delay of an individual component are assigned to a transition.

The rest of this paper is organized as follows. Section 3 describes the definition of petri net, discrete timed stochastic petri net and a method to model a hardware in petri net. In addition to this, it shows how total power and delay of hardware can be computed. Section 4 describes how the parameters such as capacitances and resistances can be extracted from the transistor level. Section 5 shows the experiments and results of our approach. Section 6 concludes the paper with remarks and possible future work.

III. Hardware modeling

A. Petri net

Petri nets consist of a set of places (P) and a set of transition (T) [6]. ”The states of the model are represented as places, the passive components” and ”The action oriented to the states of the model are represented as transitions, the active components”. The places and transitions are connected by arcs (A). The arcs are directed and can only connect a transition with a place or vice-versa. A transition is said to be enabled if there are enough token in each of the input places as specified by the arcs connecting the input places to the transition. An enabled transition can fire if the other conditions associated with the transition are satisfied.

1) Definition: A Petri net is a triplet (P, T, F),

- P is a finite set of places,
- T is finite set of transitions (P $\cap$ T = $\emptyset$),
- F $\subseteq$ (P $\times$ T) $\cup$ (T $\times$ P) is set of arcs (flow relation)
- the preset of x $\in$ P $\cup$ T denotes the set $\bullet$x = {y $\in$ P $\cup$ T $|$ (y,x) $\in$ F}, and its poset x$\bullet$ = {y $\in$ P $\cup$ T $|$ (x,y) $\in$ F}

A place is called an input place of a transition t iff there exists a directed arc from p to t. Place p is called an output place of transition t iff there exists a directed arc from t to p.

B. Discrete timed stochastic petri net

Definition: A Petri net with triplet (P, T, F) is a discrete timed stochastic petri net [6,7] if each transition t $\in$ T is associated with a distributed random variable that expresses the delay from the enabling to the firing of the transition. The distribution of delay of each transition can be represented in three forms, which are normal distribution, exponential distribution and arbitrary distribution. In this work, we chose the arbitrary distributed function for all transitions because modeling system behavior with exponential or normal distribution may not estimate the system characteristics accurately.

C. Power/delay estimation

To model the target architecture in petri net, we modified the petri net by defining one more tuple S such that S=(s_1, ..., s_n) where each element of S represents the sequence of firing of transition t. As a particular transition is fired, a token moves from one place to another place. We assume that place stores a token after a transition is fired but does not contribute to the power and performance, unlike this transition which represents a particular hardware contributes to power and delay once it is fired. Let D and P be the random variables denote delay and power consumption of a hardware component which is assigned to one of the transition t in petri net. For each such random variables D and P, let F_d: R $\rightarrow$ [0,1] and F_p: R $\rightarrow$ [0,1], denote their distribution function respectively, i.e.,

$$\mathrm{F_d(d) = Prob(D \leq d) = \sum_{i=-\infty}^{d} f_d(i)} \tag{1}$$

$$\mathrm{F_p(p) = Prob(P \leq p) = \sum_{i=-\infty}^{p} f_p(i)} \tag{2}$$

With the given density functions f_p(n) and f_d(n), sequence of firing of transition s $\in$ S, our method estimates the total power and delay of the hardware architecture by using equations (3) and (4) shown below.

$$\mathrm{f_{p_j}(i) = \sum_{j=1}^{m}\sum_{i=0}^{n} f_{p_j}(i) \cdot f_{p_j+1}(n-i)} \tag{3}$$

$$\mathrm{f_{d_j}(i) = \sum_{j=1}^{m}\sum_{i=0}^{n} f_{d_j}(i) \cdot f_{d_j+1}(n-i)} \tag{4}$$

In the above equation (3) and (4) m stands for the number of transitions t $\in$ T in the petri net based hardware architecture where each transition is fired according to the order provided by sequence S; and n is the number of discrete values of the density functions f_p(n) and f_d(n). Mathematically, total power and delay can be estimated by convolving individual density function of transitions.

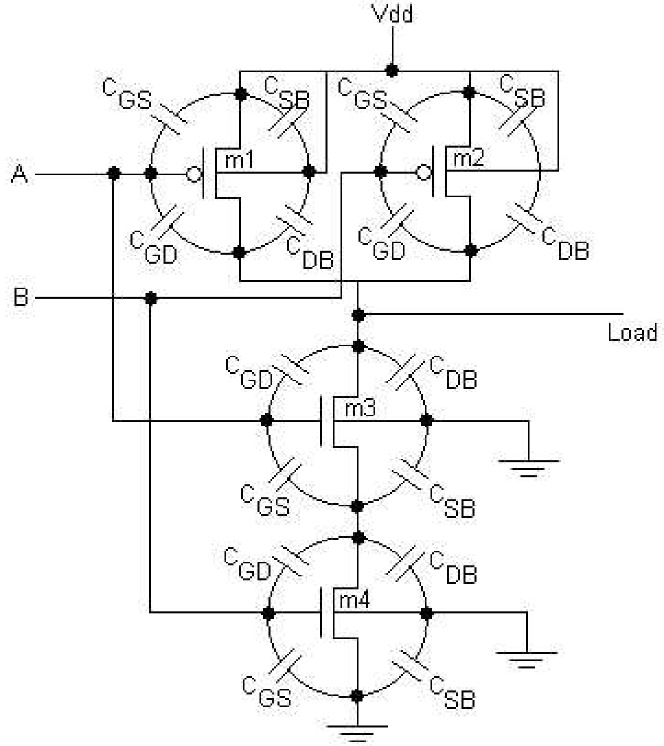

Fig. 1. Capacitances of 2-inputs NAND gate

IV. Parameter extraction

For a circuit described at the logic level there is only access to input and output nodes of a gate. Therefore, to evaluate the circuit characteristics at the gate level, all passive elements of the circuit have to be modeled as a concentrated form and represent them at their input and output nodes. In this section we describe the estimation of equivalent capacitance and resistance for any logic gate from its transistor level description. For the simplicity we are neglecting the effect of inductance on the power and delay of circuits.

A. Estimation of capacitance

The MOSFET transistor has a number of parasitic capacitances [4], which must be accounted for the circuit design: gate-source capacitance (C_{GS}), gate-drain capacitance (C_{GD}), gate-bulk capacitance (C_{GB}), source-bulk capacitance (C_{SB}) and drain-bulk capacitance (C_{DB}). In this work, we assume that capacitance due to C_{GB} does not have significant effect on the overall power and delay that's why we consider all except the C_{GB}. Although these capacitances are nonlinear function of voltage, for the general approach we assume them as a linear, time-invariant element. Given some technology parameters and the size of the transistor (W/L), a value for each of these parasitic capacitances can be computed [4]. For estimating an equivalent capacitance of gate from the transistor level, all internal nodes capacitance of that circuit are to be considered. The capacitance of the internal nodes depends on the logic value of other inputs. To model this effect we assume that each input has a certain probability of having logic value '0' or '1'. Let p_1 be the probability that the input is logic '1' and p_0 be the probability that the input is logic '0'. For an example, we estimated the capacitance of two inputs NAND gate shown in Fig. 1. In this case input A and B have four possible combinations, which are 00, 01, 10 and 11.

Capacitance at input 'A':
For input combination 00 and 10, the transistor m_4 is OFF and the equivalent capacitance at the gate terminal of m_3 is $C^{'}_{g,3}$ is,

$$C^{'}_{g,3} = p_0(C_{GD,3} + \frac{C_{GS,3} * C^{'}_{eq}}{C_{GS,3} + C^{'}_{eq}}) \tag{5}$$

$$C^{'}_{eq} = C_{SB,3} + C_{GD,4} + C_{DB,4} \tag{6}$$

For input combination 01 and 11, the transistor m_4 is ON and the equivalent capacitance at the gate terminal of m_3 is $C^{''}_{g,3}$ is,

$$C^{''}_{g,3} = p_1(C_{GD,3} + C_{GS,3}) \tag{7}$$

Then, total capacitance at gate terminal of m_3 due to four possible input combinations is $C_{g,3}$,

$$C_{g,3} = C^{'}_{g,3} + C^{''}_{g,3} \tag{8}$$

$$C_{g,3} = p_0(C_{GD,3} + \frac{C_{GS,3} * C^{'}_{eq}}{C_{GS,3} + C^{'}_{eq}}) + p_1(C_{GD,3} + C_{GS,3}) \tag{9}$$

Total capacitance at the input terminal A is,

$$\begin{aligned} C_A = p_0(C_{GD,3} + \frac{C_{GS,3} * C^{'}_{eq}}{C_{GS,3} + C^{'}_{eq}}) + \\ p_1(C_{GD,3} + C_{GS,3}) + C_{GS,1} + C_{GD,1} \end{aligned} \tag{10}$$

Capacitance at input 'B':
In a similar way, we can estimate the capacitance at input B. For input combination 00 and 01, transistor m_3 is OFF and capacitance at the gate terminal of m_4 is $C^{'}_{g,4}$,

$$C^{'}_{g,4} = p_0(C_{GS,4} + \frac{C_{GD,4} * C^{''}_{eq}}{C_{GD,4} + C^{''}_{eq}}) \tag{11}$$

$$C^{''}_{eq} = C_{SB,3} + C_{GS,3} + C_{DB,4} \tag{12}$$

For input combination 10 and 11, the m_3 transistor is ON and capacitance at the gate terminal of m_4 is $C^{''}_{g,4}$,

$$C^{''}_{g,4} = p_1(C_{GD,4} + C_{GS,4}) \tag{13}$$

Then, total capacitance at the gate terminal of m_4 with four possible input combination is $C_{g,4}$,

$$C_{g,4} = C^{'}_{g,4} + C^{''}_{g,4} \tag{14}$$

$$C_{g,4} = p_0(C_{GS,4} + \frac{C_{GD,4} * C^{''}_{eq}}{C_{GD,4} + C^{''}_{eq}}) + p_1(C_{GD,4} + C_{GS,4}) \tag{15}$$

Hence, total capacitance at input B of the NAND gate is C_B,

$$C_B = p_0(C_{GS,4} + \frac{C_{GD,4} * C''_{eq}}{C_{GD,4} + C''_{eq}}) + \\ p_1(C_{GD,4} + C_{GS,4}) + C_{GS,2} + C_{GD,2} \tag{16}$$

Capacitance at output terminal C_{Load}:,

$$C_{Load} = C_{GD,3} + C_{DB,3} + C_{GD,1} + \\ C_{DB,1} + C_{GD,2} + C_{DB,2} \tag{17}$$

Similarly, we estimated equivalent capacitance for FIFO, full adder, multiplier etc. by following the same method presented above.

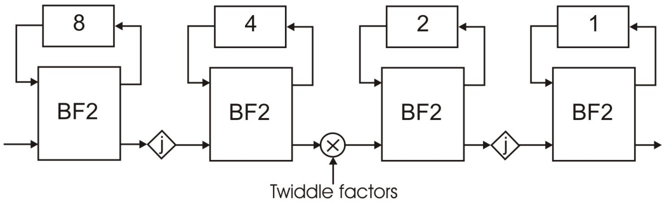

Fig. 2. FFT Hardware Architecture

B. Estimation of resistance

We estimated the equivalent resistance at the input terminals of a gate by only considering the interconnects which are input to the gate. The resistance at the output terminal of MOSFET is estimated by considering on-resistance of pMOS and nMos transistors. To estimate this resistance, we consider the longest path for charging and discharging at the output terminal. The on-resistance pMOS and nMOS transistors are R_p and R_n respectively and can be expressed as follow,

$$\mathrm{R_p} = \frac{\mathrm{V_{dd}}}{\mathrm{k_p(V_{dd} - V_{Tp})^{\alpha}}} + \frac{\mathrm{V_{dd}}}{\mathrm{k_p[2(V_{dd} - V_{Tp})V_{dd} - \frac{V_{dd}^2}{2}]}} \tag{18}$$

$$\mathrm{R_n} = \frac{\mathrm{V_{dd}}}{\mathrm{k_n(V_{dd} - V_{Tn})^{\alpha}}} + \frac{\mathrm{V_{dd}}}{\mathrm{k_n[2(V_{dd} - V_{Tn})V_{dd} - \frac{V_{dd}^2}{2}]}} \tag{19}$$

Where k_p and k_n are the gain factor of pMOS and nMOS transistors, respectively and V_{Tp} and V_{Tn} are the threshold voltage of pMOS and nMOS transistors, respectively.

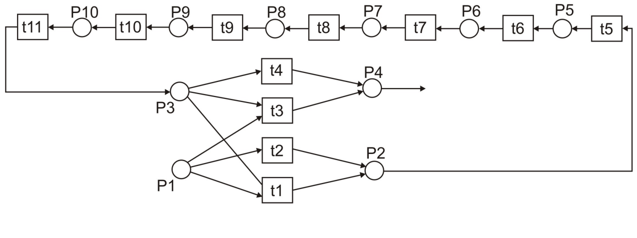

Fig. 3. Petri net model of butterfly unit and FIFO with depth 8

V. Experiment and results

To test the proposed method for high level estimation of power and delay, we chose a hardware architecture of FFT with length 16 as shown in Fig. 2. Its architecture consists of four butterfly units, four FIFOs with depth 8, 4, 2 and 1, multiplier and adder/subtracter as major units. For the petri net representation of FFT, we followed hierarchical approach by first modeling the butterfly architecture, FIFO, multiplier, adder/subtracter separately and connected them together to get model of an architecture level. As mentioned in Section 5, we extracted capacitances and resistances of logic cells from the transistor level and assigned the density functions of power and delay to each transition. Note that in this work power and delay of each individual hardware unit are assigned only to individual transition not in place. Fig. 3 shows the petri net model of butterfly and FIFO of depth 8. In this model, t_3 and t_1 transitions represent adder and subtracter respectively and t_4 and t_2 represent the multiplexer. In addition to this, transitions t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11} and places P_5, P_6, P_7, P_8, P_9, P_{10} represent the FIFO with depth 8. Similarly, we modeled other hardware components and assigned their power/delay information to the corresponding transitions of petri net model. The top level petri net model of FFT hardware with their corresponding power and delay function assigned to each transition is shown in Fig. 4. We did not present characteristics curve of all individual components but we presented some, which are significant to the power/delay characteristics. Fig. 5 and 6, are the estimated delay and power of adder considering 0.35 μ CMOS technology, respectively. The delay and power are estimated for upto 64 bit in those figures. These estimated data were later compared with the post synthesis report of the Synposys, which shows around 2% of error. This is because we avoided effect of interconnects. In the Fig. 5 and 6, the probability of having logic ’1’ or ’0’ at the input terminal of adder were chosen arbitrary. We estimated four different possible value of delay and power considering four different combination of inputs logic ’1’ or ’0’ at the input of adder. There can be more combination than the considered but for the test purpose we neglected them. We derived the density function of delay and power for 16 bit adder from the curve shown in Fig. 5, 6 and assigned their values to the transitions t_3, t_9, t_{15} and t_{21} in Fig. 4. Similarly, we assigned the density functions to all transitions of top level FFT model.

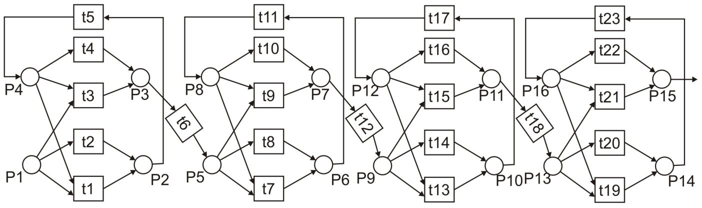

Fig. 4. Petri net model of FFT hardware architecture

We computed the total power consumption and delay of the FFT hardware architecture using equations 3 and 4,

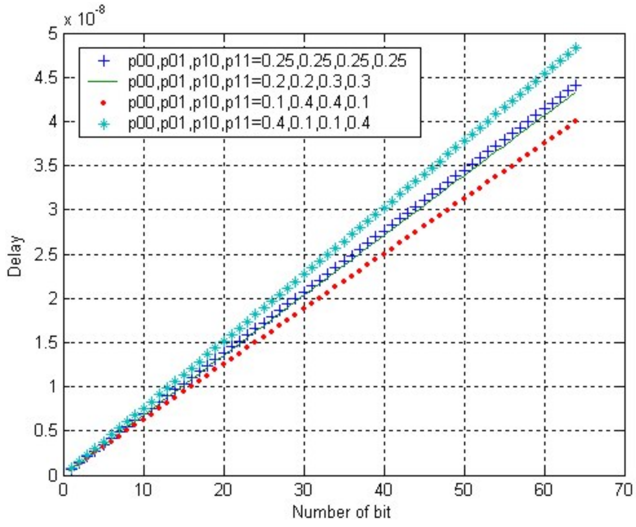

Fig. 5. Delay of full adder for different bits n

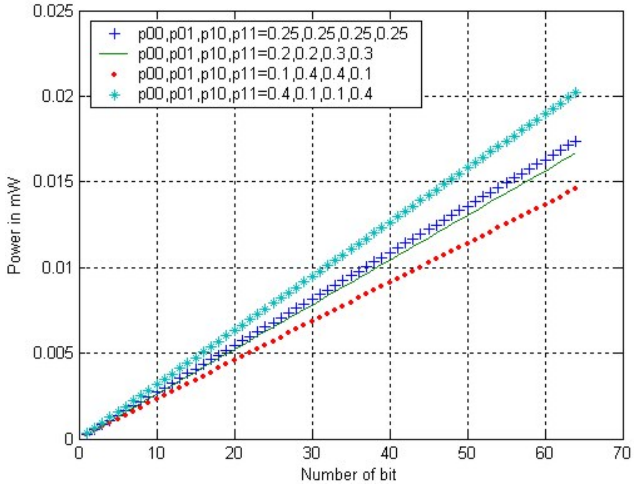

Fig. 6. Power of full adder for different bits n

respectively. These computed power and delay are shown in Fig. 7 and 8. The FFT with 16-bit adder/substractor, 16-bit wide FIFOs and 16-bit multiplier has minimum and maximum power consumption around 270 mW and 295 mW respectively. Similarly, in Fig. 8 the total delay of FFT ranges from 602 ns to 980 ns. We compared these results with the post synthesis results of the same architecture using Synopsys, the differences are 11% for power and 7% for delay from the mean value of estimated power and delay respectively. With this modeling approach, a designer can replace components such as 16-bit adder, multiplier by 32-bit adder, multiplier etc. and check quickly what is the maximum power and delay of the architecture which is to be implemented later in hardware. This gives the designer a possibility to explore design space such as finding minimum and maximum power/delay information for different design alternatives at a high level; and to map the right solution to the target architecture. However this method of modeling and estimating the parameters of the target architecture is fast and efficient, in terms of absolute accuracy it may not be so promising to the estimation using synthesis tools. But, with the synthesis tool designer should describe whole system in HDL which is of course more time taking than the approach we proposed. At high level power/delay estimation, relative accuracy of estimation is much more important than absolute accuracy, since what we really want to know is whether one alternative is better than other without describing system specification in any HDL language.

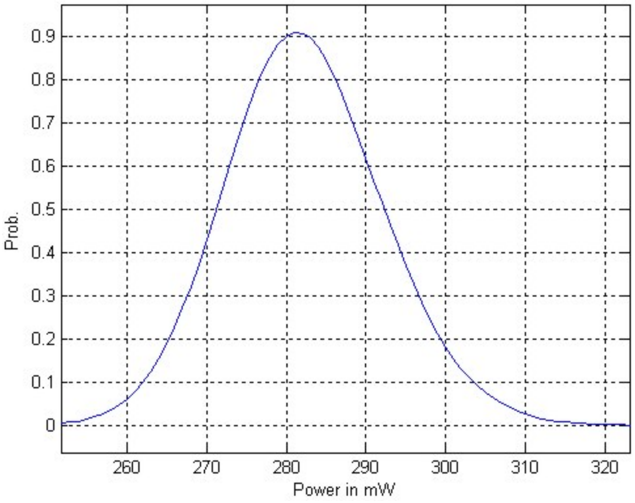

Fig. 7. Total estimated power of the FFT

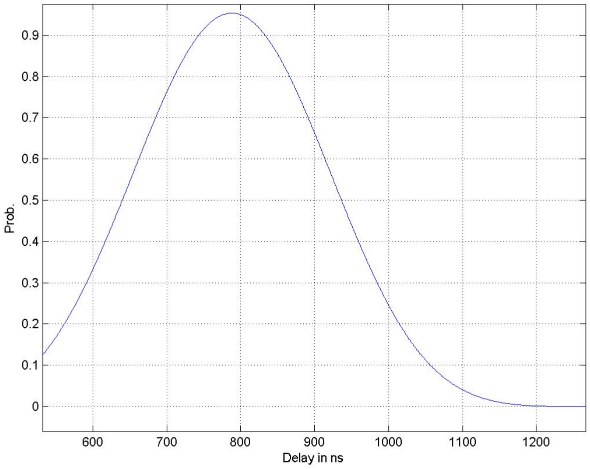

Fig. 8. Total estimated delay of the FFT

VI. Conclusion

Though the classical method of estimating power and delay are promising in terms of accuracy, they are not efficient if system to be designed is complex. We propose a method of modeling hardware architectures and estimating power/delay at the architecture level based on petri net. This method of modeling gives a designer a possibility to explore the design space and map the post partitioning hardware unit to the target architecture. The hardware is modeled in a petri net by extracting the parameters such as capacitance and resistance of gates from the transistor level. For an experimental purpose, we chose a 16 point FFT architecture and modeled it hierarchically in petri net. The total power and delay of the modeled FFT were evaluated for 16-bit adder/subtracter, multiplier, FIFO and compared their mean value of power and delay to the post synthesis report of Synopsys. The result shows an accuracy of about 91 % which shows the method is efficient to narrow the design space by early estimating power/delay; and to map to the target architecture.

Since the petri net representation is efficient for modeling and performance analysis of a complex system, we will extend our research work in future to model a system with mixed hardware and software. With this approach we can estimate the system performance to make the best decision.

REFERENCES

[1] F. Balarin, M. Chiodo, P. Giusto, H. Hsieh, A. Jureska, L. Lavagno, C. Passserone, A. Sangiovanni-Vincentelli, E. Sentovich, K. Suzuki and B. Tabbara, *"Hardware-software Co-design of embedded systems: The POLIS approach.*" Kluwer Academic Publishers, Norwell, MA., 1997.

[2] J. Henkel and R. Ernst, *"High-Level Estimation Techniques for Usage in Hardware/Software Co-Design.*" In IEEE/ACM Proc. of Asia and South Pacific Design Automation Conference, pages 353-360 February 1998.

[3] R. Henning and C. Chakrabarti, *"An approach to switching activity consideration during high-level, low-power design space exploration.*" IEEE transactions on circuits and systems-II: Analog and digital signal processing, Vol.49, No. 5, May 2002.

[4] J. P. Uyemura, *"Circuit Design for CMOS VLSI.*" Ed. Kluwer Academic Publishers, USA.

[5] F. Najm, *"A Survey of Power Estimation Techniques in VLSI Circuits.*" IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 2, Dec., pp. 446-455.

[6] T. Murata, " *Petri nets: properties, analysis and applications.*" Proc. of the IEEE, Vol. 77., No. 4, April 1989.

[7] Pieter Kritzinger with F. Bause, *"'Introduction to stochastic petri net theory."'* Published by Vieweg Verlag, Germany.

[8] J. Henkel and Y. Li, *"Avalanche: An Environment for design space exploration and optimization of low-Power embedded systems.*" IEEE Transactions on VLSI systems, Vol. 10, No. 4, Aug. 2002.

[9] K. Kurdahi, D. Gajski, C. Ramachandran, and V. Chaiyakul. *"Linking register-transfer in physical levels of design.*" IEICE Transactions on Information and Systems, Sept. 1993.

[10] N. H. E. Weste and K. Eshraghian *"Principles of CMOS VLSI design: A systems perspective second edition.*" Addison-Wesley publishing company, Oct. 1994.