Low-Power Techniques for Network Security Processors *

Yi-Ping You[1], Chun-Yen Tseng[2], Yu-Hui Huang[1],
Po-Chiun Huang[2], TingTing Hwang[1], and Sheng-Yu Hsu[3]

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan[1]
Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan[2]
Design Service Center, Industrual Technology Research Insitude, Hsinchu, Taiwan[3]

Abstract— In this paper, we present several techniques for low-power design, including a descriptor-based low-power scheduling algorithm, design of dynamic voltage generator, and dual threshold voltage assignments, for network security processors. The experiments show that the proposed methods and designs provide the opportunity for network security processors to achieve the goals of both high performance and low power.

I. Introduction

As the security issues involved in network-aware events become serious concerns of users, many security algorithms that require tons of computation are developed to provide adequate security. To deal with the large amount of data communication and computation when processing the security operations, the idea of using network security processors is brought out to provide dedicated security processing and to accelerate these processes.

In addition to high performance, network security processors are required to be low power consumption. High power consumption is translated to heat energy which makes the processor unstable and raises the cost of packaging and cooling systems. In addition, for embedded and mobile systems, the working time directly depends on the battery life. Therefore, the power dissipation becomes an important concern in the design of network security processors.

In the previous work by Su et al. [1] and Hong and Wu [2], traditional low-power techniques were used to develop the network security processor proposed by Design Technology Center at National Tsing Hua University. With the help of commercial CAD tools, power optimization techniques including macro cell mapping, control bit encoding, don't care optimizing, path balancing, gate input ordering, and transistor sizing were utilized [3]. The cryptographic modules are redesigned with modified algorithms which cost less memory buffers and smaller die size with higher performance. However, such techniques are not sufficient enough for low power considerations.

In this paper, we focus on the issue of further low-power techniques at the design and architectural phases of the network security processor. We propose several techniques, including a descriptor-based variable-voltage scheduling algorithm, dynamic voltage generations, and dual-threshold voltage techniques to provide advanced power optimizations. The variable-voltage scheduling algorithm schedules descriptors, which can be treated as tasks in network security processors, with variable operating voltage and even shuts down crypto modules when they are not in use. The dynamic voltage generator proposed a closed-loop architecture to generate supply voltage from 1.0 to 2.0 V with four levels. Finally, the dual threshold voltage assignment helps solve the problem of leakage power dissipation in the design network security processors.

The rest of this paper is organized as follows. We introduce the architecture issues of security processors in Section II. Next, we present a scheduling algorithm for descriptor-based security processor in Section III. Then, we describe voltage generation in Section IV. The dual threshold voltage technique is illustrated in Section V. Finally, concluding remarks and future work are drawn in Section VI.

II. Review of Network Security Processors

We construct a configurable architecture of descriptor-based security processor. To run this security processor, there should be a program disassembling the crypto operations into a list of descriptors. The descriptor contains the type of en-/decryption functions, the length of raw data, the key information, and the destination memory address of the result data. After settling descriptors in the memory, a trigger signal to the SP will be pulled up. Such architecture would start retrieving the descriptors, decode the crypto operations, and automatically fill data into its operation modules. When the crypto operations are finished, the result data will store back to the memory.

In our design, the architecture template (Figure1) is mainly assembled with a main controller, a DMA module, internal buses, and crypto modules. The main controller has a slave interface of external bus which accepts the control signals and returns the operation feedback via the interrupt port. Inside the main controller, there are a resource allocation module keeping the resource status table of external bus master interfaces, channels, transfer engines, internal buses, and crypto modules. The process scheduler module and power manage module are

*This work was supported in part by Ministry of Economic Affairs under Grant No. 92-EC-17-A-03-S1-0002 and Industrial Technology Research Institute under Grant No. S3-93030-4 in Taiwan.

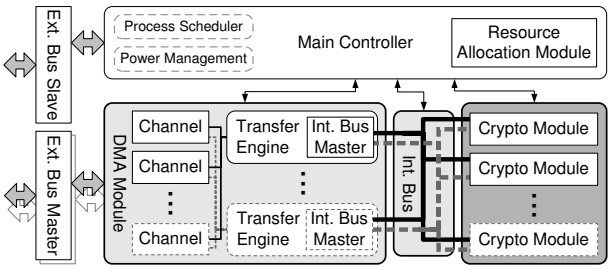

Fig. 1. Security Processor architecture template

added in the main controller for performing the features of scheduling algorithm and low power control.

The DMA module integrates master interfaces of external bus, the channels containing the descriptor information, and the transfer engines. According the data in the channel, the transfer engine requests the external bus usage and transfers the processing data from memory. After that, the transfer engine passes the data on the dedicated crypto module via the internal bus.

The internal buses are designed to support multiple layers which will be needed for high speed data transmission. The crypto modules could be any type of cryptography functions; for example, the RSA, AES, SH5, 3DES and more. The execution time of the crypto module may be different. The crypto module will return a signal to main controller, when the en-/de-crypt operations are finished.

III. Variable-Voltage Scheduling

As mentioned in Section II, the network security processor uses a descriptor-based model to proceed with encryption and decryption operations. The descriptor-based model makes it easier to distribute tasks at one's own choice: for instance, one could allocate a specific task to be run on a certain processing unit or assign the operating voltage for a task simply by descriptors. In this section, we propose a real-time variable-voltage scheduling algorithm that scales the operating voltages of crypto modules of the network security processor dynamically by scheduling descriptors and evaluate the proposed approach in the end of this section. A more generic scheduling algorithm has been proposed in [4]. We assume that the descriptor-based network security processor contains several major crypto modules (CMs) which are capable of K-level supply voltages and power-gating (PG) mode.

A. Scheduling Algorithm

The proposed scheduling algorithm is based on the well-known earliest deadline first (EDF) algorithm [5]. Figure 2 lists the algorithm. Assume there are n descriptors to be scheduled. First, we sort descriptors in ascending order by deadlines, namely $D_1, D_2, \ldots, D_n$, and put them in a list of unscheduled descriptors, i.e., the reservation list. We then extract each descriptor from the list on the basis of the schedule. Suppose the system provides m crypto modules, and each processing element is capable of K-level supply voltages, where level 1 represents the lowest voltage and level K represents the highest voltage. Steps 1–3 in Figure 2 describe these procedures.

Real-Time Variable-Voltage Scheduling Algorithm with Reservation Lists in Multiple Crypto Modules

Input: n unscheduled descriptors and m CMs
Output: Schedule of power-gating commands and the n descriptors with variable supply voltages at $CM_{1\ldots m}$

1. Sort descriptors by deadlines in ascending order; i.e., D_1, D_2, ..., D_n.
2. Put them in a list, called the *reservation list* of the target processing element CM_j. In the beginning, j=1.
3. Remove the first descriptor, namely D_i, which has the earliest deadline, from the *reservation list*. Repeat steps 3–6 while the list is not empty.
4. Compute the slack time for descriptor D_i with both the highest- and lowest-voltage pseudo schedulers; i.e., $\delta_i(V_H)$ and $\delta_i(V_L)$.
5. Compute the computation time of D_i at the highest- and lowest-voltages; i.e., $c_i(V_H)$ and $c_i(V_L)$.
6. Let O_i be the time overhead of voltage scaling for D_i.
   Schedule D_i in the following rules:
   - If $c_i(V_L) + O_i \leq \delta_i(V_L)$, schedule D_i at CM_j with V_L if possible†.
   - If $\delta_i(V_L) < c_i(V_L) + O_i \leq \delta_i(V_H)$, call the *decision algorithm*.
   - If $c_i(V_L) + O_i > \delta_i(V_H)$ and
     - if $c_i(V_H) + O_i \leq \delta_i(V_H)$, schedule D_i at CM_j with V_H.
     - if $c_i(V_H) + O_i > \delta_i(V_H)$, put D_i in a new list.
7. Insert power-gating commands at the beginning and the end of idle period for CM_j if deserving.
8. If CM_j is the last available CM and the new list is not empty, then report the possible failure of real-time scheduling.
9. If the new list is not empty, let $j = j + 1$, use the new list as the *reservation list* of the target CM_j, and go to step 3.
10. If $j < m$, then gate off the power of $CM_{j+1} \ldots CM_m$ all the time.

†Schedule D_i at V_L if deadline is met and energy overhead is deserving.

Fig. 2. Reservation-list scheduling algorithm for variable-voltage problems in multiple CMs.

For utilizing power gating capabilities, we shall try to make tasks run successively without intermission and let idle time be together because power gating mechanisms cost much more expense than DVS does in performance and power. Next, in step 4, we compute the slack time for descriptor D_i with both the highest- and lowest-voltage pseudo schedulers, denoted as $\delta_i(V_H)$ and $\delta_i(V_L)$. The slack time $\delta_i(V)$ represents the maximum time interval allowed for descriptor D_i to execute while all the remaining descriptors in the reservation list are scheduled in reverse order with supply voltage V. In step 5, we compute the computation time of descriptor D_i at both the highest- and lowest-voltages, denoted as $c_i(V_H)$ and $c_i(V_L)$. In step 6, we compare $c_i(V_H)$ and $c_i(V_L)$ with $\delta_i(V_L)$ and $\delta_i(V_H)$ to decide which voltage should be applied to the descriptor. This algorithm results in the following three possible scenarios:

(a) $c_i(V_L)$ plus time-overhead of voltage-scaling (O_i) is smaller than or equal to $\delta_i(V_L)$. If energy-overhead of voltage-scaling is less than energy-saving, we can schedule descriptor D_i at the lowest voltage without affecting any descriptor in the future because there are no overlaps between descriptor D_i and the unscheduled descriptors while those descriptors are assumed to be executed at the lowest voltage.

(b) $c_i(V_L)$ plus time-overhead of voltage-scaling is larger than $\delta_i(V_L)$ and smaller than or equal to $\delta_i(V_H)$. If this happens, we call a decision algorithm to decide at which voltage descriptor should D_i be scheduled. The decision algorithm weights the alternatives to optimize the overall costs and selects a appropriate operating voltage level for D_i, using a criterion such as the power or energy consumption.

(c) $c_i(V_L)$ plus time-overhead of voltage-scaling is larger than $\delta_i(V_H)$. This means it is impossible for descriptor D_i to complete its execution by its deadline at any voltage lower than the highest voltage, and hence we must schedule it at the highest voltage for its deadline to be met. If descriptor D_i is unschedulable for current CM, we put it in a new list that contains all unschedulable descriptors.

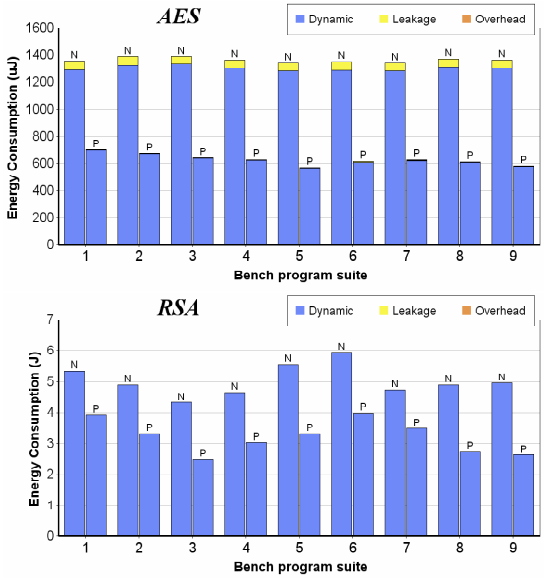

Fig. 3. Energy consumption of AES and RSA cypto modules

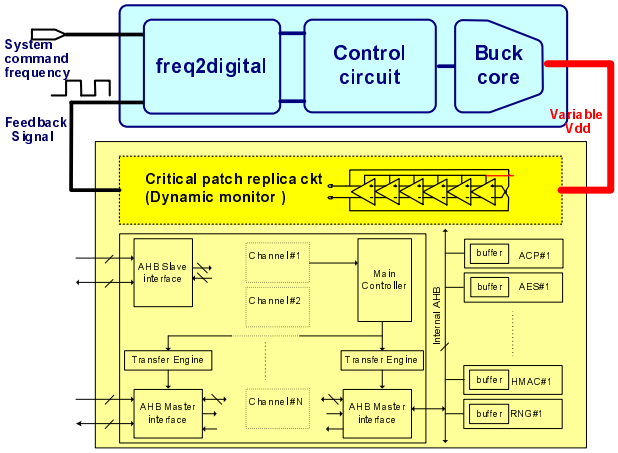

Fig. 4. The Proposed FLL-based Voltage Generation Architecture

In step 7, we check the remained idle time between the scheduled descriptors in the current CM and determine power gating commands to be inserted if they are deserving of energy-saving. In step 8 and step 9, if the new list generated in step 6 is not empty, we use the list as the reservation list for the next available CM and schedule it by the same procedures in steps 3–6. If no CM is available for scheduling, the scheduler should report the failure. At the last step, we will turn off all unused CMs by power-gating to minimize both static and dynamic power savings.

B. Experimental Results

The benchmark suites in the experiment are generated with a randomized descriptor generator which generates descriptors — the ratio of AES to RSA descriptors is 60 to 1 — in uniform, normal, or exponential distributions in terms of arrival and deadline time. The benchmark suites numbered with 1–3, 4–6, and 7–9 are generated with uniform, normal, and exponential distribution, respectively. We generated 500 distinct descriptor files for each suite and computed their average energy consumptions of different components from the results of the simulator, as shown in Figure 3. The bars labeled by N are the scheduling results without power management and others labeled by P are the results with enabling our proposed power management. The energy-overhead of applying voltage scaling and power gating is too few to be exhibited clearly on the charts, and so does the leakage of RSA modules. The left chart gives the energy consumption of AES modules, and the right chart gives that of RSA modules for all benchmark suites. The final results with latency approximation are also confirmed by the simulator that no deadline missing is reported in all benchmarks. Although the most energy consumptions are dominated by RSA operations in our experimental architecture and workload, the charts show that our scheme performs well for all components in the system. The summary in these figures present that the average energy reduction of AES and RSA modules are 54% and 36%, respectively.

IV. Dynamic Voltage Generator

The dynamic power dissipation per switching event of a CMOS circuit is given by

$$P_{dynamic} = \lambda C_{total}\, V^2\, f_{clk} \tag{1}$$

where λ is the transition factor, C_{total} is the load capacitance, V is the supply voltage, and f_{clk} is the the clock frequency.

Active-idle and frequency scaling are two common methods to get compromise between peak power dissipation and speed. But for efficient energy usage, circuit should be operated under the optimal supply voltage. In this section a voltage generator which can dynamically provide voltage level according to the decision of system scheduler is proposed. This voltage scaling technique is expected to supply the critical function block, like encryption engines in this design, with adjustable voltage rather than fixed level to achieve better power efficiency.

A. FLL-based Voltage Generation Architecture

Figure 4 shows the proposed dynamic voltage generator architecture associated with the encryption processor. The voltage generator is composed three parts: frequency to digital conversion, pulse-width modulation (PWM) controller, and bulk converter core. The two inputs, one is from system command and the other is from processor feedback, represent the target frequency that the system desires and the speed that the system now is working. These two frequencies are compared then converted as a digital code by first frequency to digital converter. This code then be modulated as PWM format in digital domain to control the on-off timing of power FETs. Bulk converter is adopted in voltage generation as it can reach higher power efficiency compared with the linear regulation approach, although it needs external LC components for filtering.

Unlike the conventional voltage downconveters or regulators, the proposed architecture is a closed-loop system. Its concept is like a frequency-locked loop (FLL) [6]. By this approach the output voltage herein is no longer a constant. Instead, the output voltage is dynamically adjusted to support processor operated at the target speed. As shown in Figure 4, a critical path replica circuit in processor is employed as the

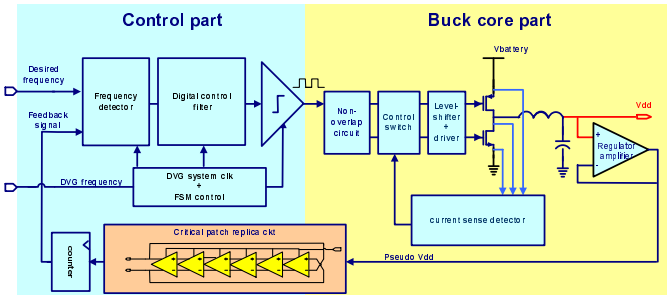

Fig. 5. Block Digram of the Proposed Dynamic Voltage Generator

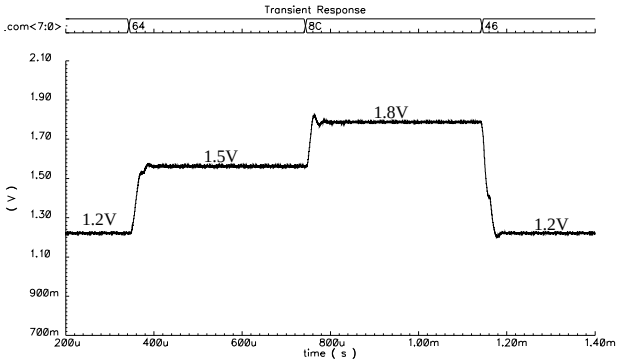

Fig. 6. Closed-loop Transient Response

speed monitoring block, which feeds back the system speed information under the voltage at the moment. After three stage processing as mentioned, the loop will generate the voltage level that provides the processor power with its speed equal to the system requirement. Compared to the fixed voltage generation, the proposed structure achieves higher yield, since the dynamically adjustable nature of closed-loop system can compensate the variations from process or environment.

B. High Efficiency Hybrid Feedback-Controlled Loop

A new design approach for high efficiency dynamic supply voltage generation is presented in Figure 5. This design introduces hybrid analog and digital signal processing to realize the PWM function with small power consumption and chip area. In conventional analog techniques, a PWM signal is created by comparing a ramp signal to a reference value. However this method requires a steady dc current flowing into an accurate capacitor to generate the voltage ramp. That will be power-hungry. To alleviate this problem, a digital PWM structure is developed. This digital PWM is achieved by counting the feedback path speed using a high speed counter. Its digital output then be used to calculate the on-off pulse width to control the switching activity of power FETs.

In the proposed digital PWM structure the power dissipation penalty is on the high speed counter. To achieve high resolution for pulse width control, the counter speed must be high. But increase counter speed unavoidably increase power. To reduce the power while keep the resolution system required, a digital dither is developed. At most four extra phases can be interpolated into the original duty cycle. With the aid of dithering technique, the operation speed on counter can be reduced. The power budget due to high speed operation demand is released therefor.

The precise control of power FETs conducting cycle is another crucial factor for power waste minimization. This is due to power leakage if either N- or P- MosFET is conducted more than expected. To keep the power FETs operated off this region, a detection and adjustment loop is constructed around these devices. It is in analog domain as the speed must be fast. Once the bulk converter internal output beyond the specified value, the detection circuits will invoke the adjustment loop to control the conducting phase, thus the leakage power can be greatly reduced.

In the proposed prototype in Figure 5 a ring-type oscillator is employed as the equivalent to the critical path delay in processor. The control signal of the ring oscillator is the power supply. This oscillator and a counter at output serve as a frequency to digital converter as the oscillating frequency is the function of supply voltage. This forms the feedback path of the proposed FLL-based voltage generation. Of course the oscillator can be replaced with real critical path in processor to monitor the real system speed.

TABLE I
PERFORMANCE SUMMARY

	Performance
Output loading	200mA
Level transition time	20μs - 30μs
Output ripple	$\leq$ 15mV
Buck core frequency	1MHz
Efficiency	91 %
Process	0.18μm CMOS

C. Experimental Result of the Proposed Voltage Generator

A FLL-based voltage generation loop is proposed to attain both software-estimated power minimization and automatic voltage adjustment for yield improvement. In this loop hybrid analog and digital signal processing can achieve high resolution while minimize the power waste. Figure 6 is the loop transient simulation result when the request is stepping up. The proposed architecture can vary the supply voltage from 1.0 to 2.0 V with four level. The transition time penalty is smaller than 30μs. Power efficiency estimated can be up to 91% with its output loading current from 50mA to 200mA.

The prototype chip uses 0.18μm CMOS technology. The performance summary is listed in Table I.

V. DUAL THRESHOLD VOLTAGE ASSIGNMENT

It is shown in [7] when the manufacturing technology scaling to 0.05 micron in the future, the static power will contribute to nearly 50% of the total power consumption. Several techniques to reduce subthreshold leakage current by multi-threshold voltage were reviewed in [8]. In dual Vth fabrication process, designers are provided with transistors that are either high threshold voltage (slow but low leakage) and low threshold voltage (fast but high leakage) by only an extra mask

```
1 Algorithm SE-DVA ()
2 Input : Graph G = (V, E)
3 Output : Set of high-Vth gates, HList
4
5  For each node v, v∈V
6    If SL(v)≥DiffD(v) {
7      Add node v to SList by its SL(v);
8      AllD(v) = 0;}
9  Construct G_max of G_s;
10 ε = SL_max − SL_max−1
11 While (ε≠0 and SList≠∅) {
12   /* Find MWIS in G_max */
13   Compute W(v) for each node v in G_max
14   While G_max≠∅ {
15     Select node v with maximum W(v) to MWIS;
16     Remove v and its neighbors in G_max}
17   For all node v in MWIS do {
18     εε = min(ε, DiffD(v) − AllD(v));
19     AllD(v) = AllD(v) + εε ;
20     If (AllD(v) = DiffD(v)) then
21       Move node v from SList to HList;}
22   Update AT/RT/SL/ε for the graph G;}
```

Fig. 7. Single Execution of Dual-Vth Assignment

layer. In this section, we will describe our dual-Vth assignment algorithm [9]. Our algorithm is inspired by the maximum independent set based slack assignment (MISA) proposed in [10].

A. MISA-based Assignment Algorithm

The pseudo-code of our single execution of dual-Vth assignment algorithm is shown in Figure 7. Given the arrival time (AT), required time (RT), and slack (SL) for each node, first, we construct a $SList(G)$ by $SL(v), v{\in}V$ in decreasing order. Note that for each node we compute the delay difference of the node working at low-Vth and high-Vth, $DiffD(v)$. If the slack is less than the difference, we will not put it into the candidate list, $SList$, for selection. The reason is that, in this situation, it is impossible to swap this node from low-Vth to high-Vth even if all slack is allocated to the node. Before we proceed the algorithm, we set the current allocated delay, $AllD(v)$ to be zero, for all nodes in $SList$.

We first select the nodes which have the maximum slack and construct a transitive closure graph, G_{max}, among these nodes. Then, we will find the maximum independent set in G_{max}. Since the maximum independent set problem is NP-complete on general graph. We propose a heuristic to solve it.

First, we assign a weight to each node in the graph and then find the maximum weight independent set in the graph. The weight of a node v, $W(v)$, is defined to reflect the gain of selecting a node to work at high-Vth:

$$W(v) = \alpha * PW(v) + \beta * UW(v) + \gamma * LW(v, dist) \quad (2)$$

The first term, PW, is defined to represent the effective power saving when $\varepsilon\varepsilon$ slack is used. It is:

$$PW(v) = \frac{\triangle P(v)}{\varepsilon\varepsilon \times (1 + Adj(v))} \quad (3)$$

where $\triangle P(v)$ is the power saving of node v when node v works at high-Vth, $\varepsilon\varepsilon$ is $min(\varepsilon, (DiffD(v) - AllD(v))$, and $Adj(v)$ denotes the number adjacent nodes to v in G_{max}. Note that $\varepsilon = SL_{max} - SL_{max-1}$.

The second term, $UW(v)$, in (2) is used to reflect how urgent of node v to be selected.

$$UW(v) = \frac{AllD(v)}{DiffD(v)}. \quad (4)$$

In this equation, $AllD(v), DiffD(v)$ are the additional delay allocated to node v and the delay difference of v when working at high-Vth and low-Vth, respectively. The closer the delay for v to work at high Vth, the urgent the node will be selected.

The third term, $LW(v, dist)$, in (2) is used to reflect the location information of the circuit. Since our proposed algorithm is called after the circuit has been placed. We can take cell location in the floorplan into account. When we select nodes to work at high-Vth, we can assign the nodes which have the most neighbors working at high-Vth a higher weight. This will bring nodes with high-Vth or low-Vth clustered in a region so as to have higher yield during fabrication process. $LW(v, dist)$ is defined as follows.

$$LW(v, dist) = hnode \quad (5)$$

where $hnode$ is the number of nodes working at high-Vth that are within distance of $dist$ from v. $dist$ is a user defined parameter. α, β, γ in (2) are the parameter to control the importance of the three terms.

After the weight of node in G_{max} is computed, the following heuristic is used to find the maximum weight independent set (MWIS). The heuristic begins with selecting the node v with maximum $W(v)$. Then node v is added to MWIS and v and all its neighbors are deleted from G_{max}. This process repeats until $G_{max} = \phi$. After finding the MWIS, $\varepsilon = SL_{max} - SL_{max-1}$ is computed, and a small delay of constant $\varepsilon\varepsilon = min(\varepsilon, (DiffD(v) - AllD(v)))$ is added to $AllD(v)$ for each node v in MWIS. If the delay added to node v is enough for it to work at high-Vth (i.e., $AllD(v) = DiffD(v)$), we will select this node to work at high-Vth and add this node to $HList$, where $HList$ is a list that store selected nodes that can work at high-Vth. If the current allocated delay of the selected node is not high enough to work at high-Vth, the node will remain in the list of $SList$. Finally, the timing information is updated for each sensitive nodes. This procedure repeats until $SList = \phi$ or $SL(v) = 0$, for all nodes in $SList$.

B. Experiment Flow for Dual Threshold Voltage

Our experiment flow will follows the flow suggested by *TSMC Multiple Vt Flow for Leakage Power Reduction version 4.0* as shown in Figure 8 . In this experiment, TSMC 0.13μm lib was used.

At the beginning, an input design is synthesized using only low-Vth library by Design Compiler. Then this synthesized gate-level Verilog code is placed and routed using Astro. After finishing placement and routing, we use PrimeTime to output the timing information file, called SDF. In this file, we can get the accurate wire delay and cell delay. With this timing information, we calculate the arrival/required time of the circuit and analyze the timing slack. Then, dual threshold voltage assignment is performed. In our case, our $Dual\text{-}Vth\ Assignment$

TABLE II
CIRCUIT DESCRIPTIONS

Cir.	CN	Characteristics
TOP	463	An Alarm Clock
MAC	2425	Multiplier and Accumulator
GCC	8204	Gravity Center Calculator
RSA	14815	Asymmetric Crypto-Processor
AES	16824	Advanced Encryption Core

TABLE III
RESULT OF TSMC AND OUR ALGORITHM

Cir.	BP	Swap Rate %			Power Reduction %		
		TSMC	Ours	Imp.	TSMC	Ours	Imp.
TOP	5.1	66.1	93.7	27.6	38.0	80.4	42.4
MAC	37.4	79.1	96.7	17.6	60.0	81.1	21.1
GCC	97.1	80.6	86.3	5.7	47.8	59.0	11.2
RSA	201.0	94.6	98.3	3.6	78.3	83.8	5.6
AES	228.7	89.3	90.2	6.3	67.6	77.4	9.8
Avg.		80.9	93.0	12.2	58.4	76.4	18.0

algorithm implemented in C is called. In the flow suggested by TSMC, the following algorithm is performed.
First, a task checks if there is any timing violation path. Then, the following iterations start to swap cell.

Iteration(1) Swap all cells not in violation path to HVT.
Iteration(1+m) Swap cells in violation path to NVT.
Iteration(1+m+n) Swap cells in violation path to LVT.

After executing the dual threshold assignment algorithm, a Verilog code using cells from low-Vth and high-Vth library is produced. Finally, this Verilog code is used to instruct Astro to do ECO placement.

To demonstrate the effectiveness of our algorithm, 5 designs were selected for benchmarking process. The descriptions of these designs are shown in Table II. The columns labeled **CN**, and **Characteristics** are the cell number of the design, and the characteristics of the design, respectively.

The comparison results are shown in Table III. The column labeled **BP** is the original leakage power consumption in nanowatt (nW) of the circuit calculated by Design Compiler. The column labeled **Swap Rate** is the ratio of the swapped number of cells to the total number cells. The cells were swapped from low to high Vt. The column labeled **Power Reduction** is calculated as $\frac{dual-Vth\ power}{original\ power}$. The column labeled **TSMC** is the results of the circuit using TSMC tool. The column labeled **Ours** is the results using our algorithm. The column labeled **Imp.** is calculated as the subtraction of **Ours** and **TSMC**. It can be seen that out method can reduce on the average 18.0% more leakage power than TSMC's.

VI. CONCLUSIONS

In this paper, we have proposed several low-power techniques for network security processors, including a descriptor-based low-power scheduling algorithm, design of dynamic voltage generator, and dual threshold voltage assignments. The experiments have shown that the proposed designs are efficient in power reduction for network security processors.

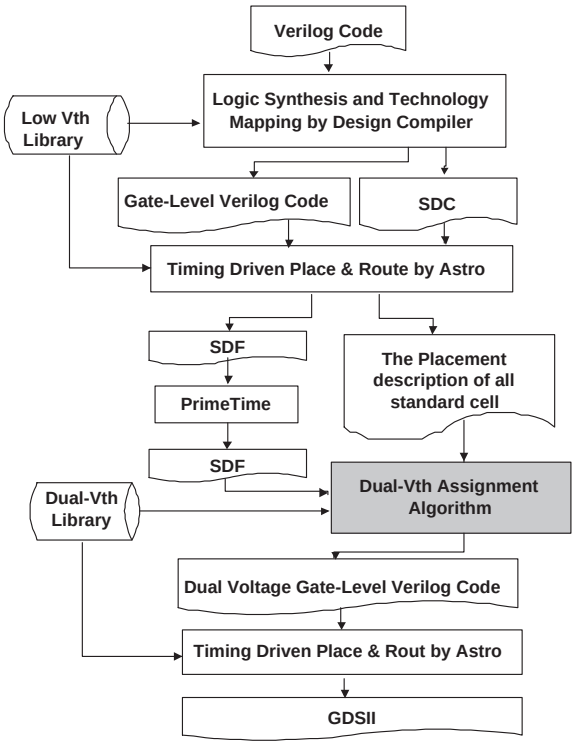

Fig. 8. Design Flow for Dual-Vth

REFERENCES

[1] C.-P. Su, T.-F. Lin, C.-T. Huang, and C.-W. Wu, "A high-throughput low-cost AES processor," *IEEE Communications Magazine*, vol. 41, no. 12, pp. 86–91, Dec. 2003.

[2] J.-H. Hong and C.-W. Wu, "Cellular array modular multiplier for the RSA public-key cryptosystem based on modified Booth's algorithm," *IEEE Transactions on VLSI Systems*, vol. 11, no. 3, pp. 474–484, June 2003.

[3] S. Devadas and S. Malik, "A survey of optimization techniques targeting low power VLSI circuits," In *Proceedings of the 32nd ACM/IEEE Conference on Design Automation*, pp. 242–247, 1995.

[4] Y.-C. Lin, Y.-P. You, C.-W. Huang, J. K. Lee, W.-K. Shih, and T.T. Hwang, "Power-aware scheduling for parallel security processors with analytical models," Submitted to *The 17th International Workshop on Languages and Compilers for Parallel Computing*, 2004.

[5] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard read-time environment," *Journal of the ACM*, 20(1):46–61, 1973.

[6] A. Stratakos, S. Sanders, and R. Brodersen, "A low-voltage CMOS DCDC converter for a portable battery-operated system," In *Proceedings of IEEE Power Electronics Specialists Conference*, pp. 619–626, 1994.

[7] S. Borkar, "Low power challenges for the decade," In *Proceedings of ASP-DAC*, 2001.

[8] J. Kao, S. Narendra and A. Chandrakasan "Subthreshold leakage modeling and reduction techniques," In *Proceddings of ICCAD*, pp. 141-148, 2002.

[9] Y.-T. Ho, T.T. Hwang, "Low power design using dual threshold voltage," In *Proceedings of ASP-DAC*, pp. 205-208,2004.

[10] C. Chen, X. Yang and M. Sarrafzadeh, "Potential slack: an effective metric of combinational circuit performance," In *Proceedings of ICCAD*, pp.198-201, 2000.