Low-Power Design Methodology for Module-wise Dynamic Voltage and Frequency Scaling with Dynamic De-skewing Systems

Takeshi Kitahara, Hiroyuki Hara, Shinichiro Shiratake, Yoshiki Tsukiboshi*,
Tomoyuki Yoda, Tetsuaki Utsumi, and Fumihiro Minami

TOSHIBA Corporation Semiconductor Company *TOSHIBA Microelectronics Corporation
580-1, Horikawa-cho, Saiwai-ku, Kawasaki, 212-8520, JAPAN
kitahara@dad.eec.toshiba.co.jp

Abstract - This paper discusses design methodology for a module-wise dynamic voltage and frequency scaling(DVFS) technique which adjusts the supply voltage for a module appropriately to reduce the power dissipation. A circuit is able to work even when the supply voltage is in transition, by using our dynamic de-skewing system(DDS). We propose a novel clock design methodology to minimize the inter-module clock skew for solving one of the major design issues in the module-wise DVFS. We also describe a method of determining the minimum supply voltage value for a module. We lead the issue to a problem of solving simultaneous polynomial inequalities. Our experimental results show that the module-wise DVFS can reduce 53% power compared with the chip-wise DVFS, and 17% more reduction was achieved by applying the minimum supply voltage proposed.

I. Introduction

As the clock frequency has become higher, reducing the power dissipation has become a more serious issue. Chips for portable electric appliances are required to reduce the power in order to make the battery life longer, and chips for high-end digital systems should be cared about issues of the heat and the packaging cost. Recently, the low power design is also an important topic from the ecological point of view.

The DVFS technique controls the supply voltage and the clock frequency dynamically, according to the performance requirement. DVFS makes a deep impact on the dynamic power reduction, because the energy consumption of a chip is nearly proportional to the square of the supply voltage. In many cases, DVFS is employed for a whole chip, that is the supply voltage and the clock frequency are changed for all of the digital parts on a chip, such as Intel XScale$^{(R)}$ technology and Transmeta LongRun^{TM} technology[1]. Recently, however, the technique to apply DVFS to the module on a chip has been proposed[2-4]. It is effective for reducing the power dissipation, as the possibility of lowering the supply voltage increases, compared with DVFS which is applied to a whole chip. When controlling a whole chip, the supply voltage and the clock frequency cannot be lowered, if there exists only one module which requires high performance[2].

Lately, the leakage power dissipation has been increasing. Its reduction is a major subject to be tackled under the 90nm technology era and beyond, as lowering the supply voltage makes the threshold voltage lower. The dynamic power dissipation decreases according to the supply voltage with the quadratic order when lowering the supply voltage; however, the sub-threshold leakage power increases with the exponential order when lowering the threshold voltage [5]. So, the ratio of the leakage power to the dynamic power in the total power dissipation is increasing, nowadays. Some papers proposed the technique of reducing the sum of the dynamic power dissipation and the leakage power dissipation by DVFS [6-8]. Multi-threshold CMOS (MT-CMOS), selective multi-threshold CMOS (Selective-MT) and variable threshold CMOS (VT-CMOS)[9-12] are the effective techniques to reduce the leakage power dissipation. The techniques can be applied, when a circuit or a module is on standby. The technique to apply DVFS to the module is also effective to reduce the leakage power, because the standby mode can be defined for each module, in accordance with its functionality. The standby time assigned to the module is longer than the one assigned to the whole chip. The module-wise DVFS has an effect not only on the dynamic power reduction, but also on the leakage power reduction.

The H.264/MPEG-4 codec LSI with the module-wise DVFS was developed, and its effectiveness was verified[2]. One feature of this LSI is that it operates even when the supply voltage is in transition. Fig.1 shows the way how the supply voltage and the clock frequency change. This mechanism is realized by using DDS[2]. In [13], the voltage transition overhead is considered. The optimal scheduler is proposed in a discretely variable voltage environment. It is assumed that no instructions in the task are executed during the transition interval; however, instructions can be executed during the interval by using DDS. Neither voltage transition overhead nor performance degradation happens.

The clock delay in a DVFS module changes continuously, when the supply voltage is in transition. On the other hand, the clock delay does not change for a module whose supply voltage is fixed. In this situation, the clock skew between a DVFS module and a non-DVFS module may become larger,

0-7803-9451-8/06/$20.00 ©2006 IEEE.

and a larger clock skew causes the timing violation. DDS can minimize the clock skew even during the transition interval, and novel methodology of generating the clock tree is required to minimize the clock skew all the time. In this paper, we propose design methodology for the module-wise DVFS. Novel strategy of the clock tree synthesis is presented for a circuit with DDS, and a method of determining the minimum supply voltage is also introduced. The remainder of the paper is organized as follows. Section II explains the module-wise DVFS using DDS. A delay control circuit(DCC), which controls the clock-delay for a DVFS module, is also described. Clock design methodology for a circuit using DDS, is given in Section III. Section IV presents conditions of the timing closure. Experimental results are shown in Section V.

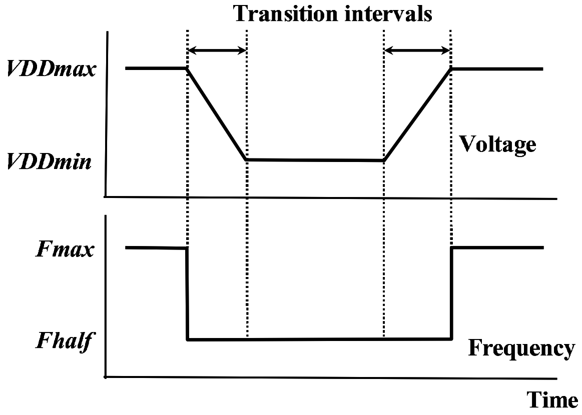

Fig.1. Behavior of voltage and frequency

II. Module-wise DVFS

The module-wise DVFS is a low power technique operating a module at an appropriate frequency and an adequate supply voltage, by using a frequency gear and a supply voltage controller. The frequency change is performed instantly; however the supply voltage transition with high accuracy and high speed is difficult. Dynamic skew control for the DVFS module, such as DDS, is necessary, so as not to interrupt the processing for the DVFS module during the supply voltage transition.

A. Dynamic De-skewing System(DDS)

DDS is introduced to realize the module-wise DVFS. Fig.2 shows the structure of DDS with the DVFS module and the non-DVFS module. In Fig.2, a variable voltage value is supplied for the DVFS module, and a fixed voltage value is supplied for the non-DVFS module. The clock signal is connected directly to the non-DVFS module, and it is connected to the DVFS module via DCC. The function of DCC is to control the clock delay to minimize the clock skew between the DVFS module and the non-DVFS module. Details of DCC are explained in the following subsection. The clock signal of the DVFS module and that of the non-DVFS module have paths which reach DCC. They are shown as the return signal A and B in Fig.2. The delay of the return signal A is adjusted to that of B. DDS measures the skew between the two return signals, and adjusts the clock delay for the DVFS module in each clock cycle. According to the measured skew, an adequate delay is added to the clock signal for the DVFS module in the next cycle.

B. Delay Control Circuit(DCC)

DCC consists of one skew measure unit and two delay generation units, as shown in Fig.2. Fig.3(a) shows the schematic of the delay generation unit A. The port CKI is connected to the root clock signal ROOTCLK, and the adjusted clock signal is supplied to the DVFS module from the port CKO. The delay of each component Ai is designed to be same for $i \in \{1,2,...,n\}$. (In Fig.3(a), the case of n=5 is shown.) The signal CKI is connected to the n switches, and

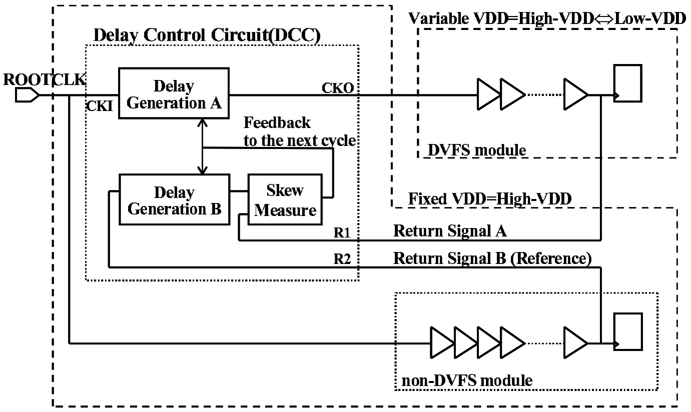

Fig.2. Dynamic de-skewing system structure

they are controlled by skew indication signals, which are described in the following explanation. Only one of the skew indication signals is set to 'H', so only one path from the signal CKI to CKO is activated when CKI changes from 'L' to 'H'. As the delay of the component Ai can be defined as a unit delay, the delay of the signal from CKI to CKO can be controlled by the step of the unit delay. For example, 3 unit delays are added in Fig.3(a).

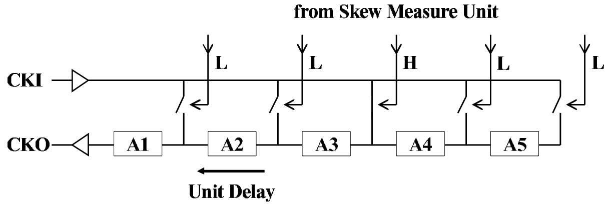

Fig.3(a). Delay generation unit A

Fig.3(b) shows the schematic of the skew measure unit and the delay generation unit B. It has two input ports. The port R1 is connected to the return signal A, and the port R2 is connected to the return signal B. It measures the time between the rise edge of the signal R1 and that of R2, as follows. In Fig.3(b), the delay of each component Bi and Ci is same as that of Ai in Fig.3(a) for $i \in \{1,2,...,n\}$. The case of n=5 is shown in Figure 3(b). If the signal R1 changes from 'L' to 'H', the output value of B1 changes from 'H' to 'L' after 1 unit delay, and that of Bi(i=2,3,...,n) changes from 'H' to 'L' one after another. When the signal R2 changes from 'L' to 'H', each output value of Bi(i=1,2,...,n) is captured by each transparent latch Li(i=1,2,...,n) after some unit delays which is determined by the delay generation unit B. There exists some m such that the value of Li($1 \leq i \leq m$) is equal to 'L' and that of Li($m < i \leq n$) is equal to 'H'. For example, m=3 in Fig.3(b). The skew measure unit has the skew indication signals. They are generated by the NOR gates whose inputs are Li(i=1,2,...,n), and only one of them is set to 'H', according to the skew between the delay of the input R1 and that of the signal R2T in Fig.3(b). According to the skew between the two, the clock delay from CKI to CKO is adjusted in the next cycle. When the supply voltage of the DVFS module and that of the non-DVFS module are same, the clock delay inside DCC is the largest. As the supply voltage of the DVFS module goes down, the clock delay becomes smaller.

III. Clock Design Methodology

This section shows the clock design methodology for a circuit including DDS. We assume that the clock tree structure is employed for the clock design.

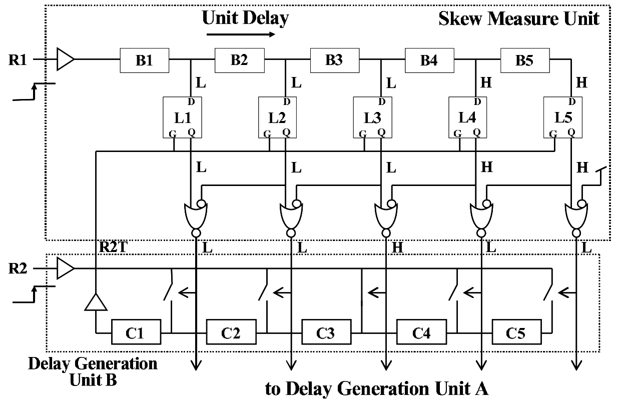

Fig.3(b). Skew measure unit and delay generation unit B

A. Problem Formulation

When designing the clock, the Clock Tree Synthesis(CTS) technique is commonly employed. Since CTS can systematically minimize the clock skew, it is helpful in making the clock design task easier and reducing the design turn around time. Clock design of DDS is complicated, because there are feedback clock signals to DCC as shown in Fig.2. The conventional CTS technique is not constructed on the assumption with feedback loops on clocks. So, a novel CTS technique is required to design a chip with the module-wise DVFS effectively. Determining the supply voltage value for the lower clock frequency is another thing to be considered. We want to let the supply voltage value as lower as possible, for reducing the power dissipation of a chip.

Our overall goal is the following. (1)Constructing novel clock design methodology which employs the CTS technique for the module-wise DVFS. (2)Clarifying the minimum clock skew the methodology can achieve. The setup and the hold timing constraints are examined with the clock skew taken into consideration. (3)Determining the lowest supply voltage which guarantees a chip to work correctly.

B. Clock Design Strategy

Fig.4 shows the clock network of DDS, and we present the clock design strategy to generate a clock tree structure for it. Let *Fmax* be the clock frequency of the clock signal ROOTCLK. The clock signal having a half of the frequency (*Fhalf*) is generated via D-type flip-flop(*F/F*). We assume that the frequency of the clock signal which is connected to DCC, switches between *Fmax* and *Fhalf* according to the performance requirement as explained in Fig.1. The frequency of the clock signal which is connected to the non-DVFS module is always *Fmax*. This design style is commonly employed in a practical design. For example, in the audiovisual chip design[2], the audio module works with

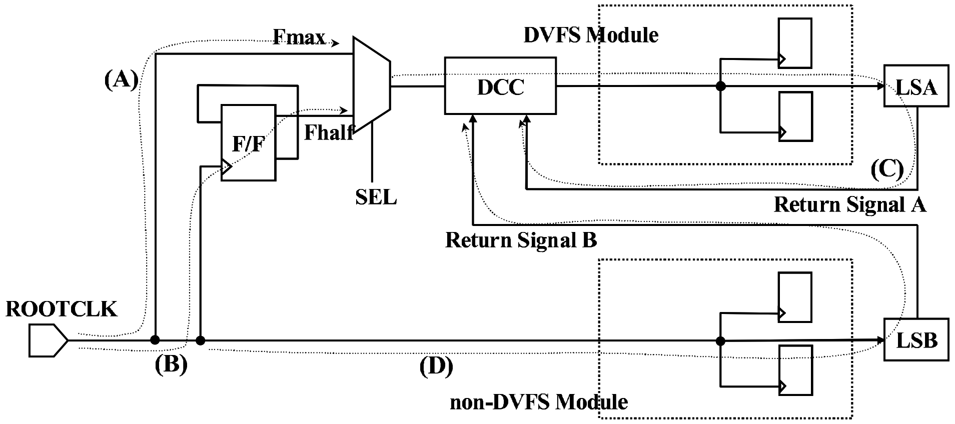

Fig.4. Clock network of Dynamic de-skewing system(DDS)

180/90MHz, because the workload of it tends to be small in many applications, while the video modules always work with 180MHz. With the exception of the clock which generates a half of frequency, Fig.4 is almost same as Fig.2. The following is the flow of generating a clock tree structure for the clock network shown in Fig.4.

III. Clock Design Methodology

This section shows the clock design methodology for a circuit including DDS. We assume that the clock tree structure is employed for the clock design.

A. Problem Formulation

When designing the clock, the Clock Tree Synthesis(CTS) technique is commonly employed. Since CTS can systematically minimize the clock skew, it is helpful in making the clock design task easier and reducing the design turn around time. Clock design of DDS is complicated, because there are feedback clock signals to DCC as shown in Fig.2. The conventional CTS technique is not constructed on the assumption with feedback loops on clocks. So, a novel CTS technique is required to design a chip with the module-wise DVFS effectively. Determining the supply voltage value for the lower clock frequency is another thing to be considered. We want to let the supply voltage value as lower as possible, for reducing the power dissipation of a chip.

Our overall goal is the following. (1)Constructing novel clock design methodology which employs the CTS technique for the module-wise DVFS. (2)Clarifying the minimum clock skew the methodology can achieve. The setup and the hold timing constraints are examined with the clock skew taken into consideration. (3)Determining the lowest supply voltage which guarantees a chip to work correctly.

B. Clock Design Strategy

Fig.4 shows the clock network of DDS, and we present the clock design strategy to generate a clock tree structure for it. Let *Fmax* be the clock frequency of the clock signal ROOTCLK. The clock signal having a half of the frequency (*Fhalf*) is generated via D-type flip-flop(*F/F*). We assume that the frequency of the clock signal which is connected to DCC, switches between *Fmax* and *Fhalf* according to the performance requirement as explained in Fig.1. The frequency of the clock signal which is connected to the non-DVFS module is always *Fmax*. This design style is commonly employed in a practical design. For example, in the audiovisual chip design[2], the audio module works with 180/90MHz, because the workload of it tends to be small in many applications, while the video modules always work with 180MHz. With the exception of the clock which generates a half of frequency, Fig.4 is almost same as Fig.2. The following is the flow of generating a clock tree structure for the clock network shown in Fig.4.

Step 1: To minimize the skew between the delay of the following two clock paths, delay buffers are inserted into the clock paths:
(A) the direct path from ROOTCLK to the multiplexer,
(B) the path from ROOTCLK to the multiplexer via D-type flip-flop.
Let $Delay_{Fmax}$ and $Delay_{Fhalf}$ be the delay of the path (A) and (B), respectively, and α be the minimum delay of a delay buffer among all delay buffers in a cell library. Then, we have:

$$| Delay_{Fmax} - Delay_{Fhalf} | < \alpha \,. \tag{1}$$

Step 2: The conventional CTS technique is applied to all flip-flops in both the DVFS module and the non-DVFS module with assigning the root as ROOTCLK in Fig.4. Clock buffers are added to the clock paths in the DVFS module and the non-DVFS module, in order to construct a clock tree structure. Let β be the maximum clock skew among all of the flip-flops, and $Delay_{FFi}$ be the clock delay between ROOTCLK and each flip-flop *FFi*. Then we have the following inequality, for any pair (*i*, *j*) such that *FFi* and *FFj* are flip-flops in the DVFS module or the non-DVFS module, under a fixed supply voltage:

$$|\, Delay_{FFi} - Delay_{FFj} \,| \leq \beta . \qquad (2)$$

Remark 1: When the supply voltage value for the DVFS module is equal to the scaling upper limit(i.e. *VDDmax* in Figure 1), the path delay from CKI to CKO inside the delay control unit A shown in Figure 3(a) is the maximum, that is the path A$n \rightarrow$ An–1... $\rightarrow$ A1 is activated. The clock delay inside the DVFS module is the smallest when the supply voltage value is the upper limit, so the largest path delay is required for the delay control unit A, to minimize the clock skew between a flip-flop in the DVFS module and one in the non-DVFS module.

Remark2: The supply voltage of the DVFS module is lower than that of DCC in some periods. So, it is necessary to add a level shifter to the outputs of the DVFS module [14]. The function of the level shifter is to lift the output voltage, when the output signal value is equal to 'H'. The level shifter is shown as LSA in Fig.4. The level shifter LSB is also added to the output of the non-DVFS module. These level shifters are treated as the leaves when the CTS is applied. So, the skew among these two and all the flip-flops is minimized, and the difference between any two of them is smaller than β.

Let γ be the unit delay in DCC, which is the delay of A*i*. Let *VDDmax* and *VDDmin* be the maximum and the minimum voltage scaling value for the DVFS module. When the supply voltage value changes between *VDDmax* and *VDDmin*, the maximum clock skew caused by DCC is γ. So, we have the following inequality, for any pair (*i*, *j*) such that *FFi* and *FFj* are flip-flops in the DVFS module or the non-DVFS module, under any supply voltage value *V* such that *VDDmin* $\leq V \leq$ *VDDmax* for the DVFS module:

$$|\, Delay_{FFi} - Delay_{FFj} \,| \leq \beta + \gamma . \qquad (3)$$

But, to be precise, only the clock skew between a flip-flop in the DVFS module and one in the non-DVFS module is affected by the delay control unit. If *FFi* and *FFj* are in the same module, there happens no clock skew penalty caused by DCC. So, we have the following inequality, for any pair (*i*, *j*) such that *FFi* and *FFj* are flip-flops in the DVFS module, under any supply voltage value *V* such that *VDDmin* $\leq V \leq$ *VDDmax* for the DVFS module:

$$|\, Delay_{FFi} - Delay_{FFj} \,| \leq \beta . \qquad (4)$$

We also have (4) for any pair (*i*, *j*) such that *FFi* and *FFj* are flip-flops in the non-DVFS module. When comparing the rising edge of R1 and that of R2, the chip was designed so that the former is always earlier. Supposing that the difference of the above two is $n * \gamma + \gamma'$, where $n \in \mathbf{N}$ ($\mathbf{N}$ is the set of natural numbers) and $0 \leq \gamma' < \gamma$, γ' is rounded off by $n*\gamma$. So, we have the following two asymmetrical inequalities, for any pair (*i*, *j*) such that *FFi* is a flip-flop in the DVFS module and *FFj* is one in the non-DVFS module, under any supply voltage value *V* such that *VDDmin* $\leq V \leq$ *VDDmax* for the DVFS module:

$$Delay_{FFi} - Delay_{FFj} \leq \beta , \qquad (5)$$

$$Delay_{FFj} - Delay_{FFi} \leq \beta + \gamma . \qquad (6)$$

Step 3: To equalize the delay of the return signal A with that of B, delay buffers are inserted into one of the above two. Let $Delay_{RA}$ and $Delay_{RB}$ be the delay of the return signal A and B, respectively, then we have:

$$|\, Delay_{RA} - Delay_{RB} \,| < \alpha , \qquad (7)$$

as α is the minimum delay of a delay buffer among all delay buffers in a cell library. Here, we observe the delay difference between the following two clock paths.

(C)ROOTCLK $\rightarrow$ multiplexer $\rightarrow$ DCC
$\rightarrow$ DVFS module $\rightarrow$ LSA $\rightarrow$ return signal A $\rightarrow$ DCC,
(D)ROOTCLK $\rightarrow$ non-DVFS module $\rightarrow$ LSB
$\rightarrow$ return signal B $\rightarrow$ DCC.

Let $Delay_C$ and $Delay_D$ be the delay of the paths (C) and (D), respectively. We can reach the following inequality by (1) and (7):

$$|\, Delay_C - Delay_D \,| < 2 * \alpha . \qquad (8)$$

Let us remember that the clock delay difference inside the DVFS module and the non- DVFS module can be ignored, as its effect is already considered in (5) and (6). The difference expressed in (8) can be reduced by the following Step 4, and we have the following inequality, under the condition that the clock frequency value *F* is equal to *Fmax* or *Fhalf*:

$$|\, Delay_C - Delay_D \,| < \alpha . \qquad (9)$$

Step 4: When the following two inequalities are both satisfied, the delay buffer whose delay is equal to α is inserted into the return signal A:

$$0 < Delay_{Fhalf} - Delay_{Fmax} < \alpha , \qquad (10)$$

$$0 < Delay_D - Delay_C < \alpha . \qquad (11)$$

It can be also realized by removing the one from the return signal B if exists, instead of the insertion. When the following two inequalities are both satisfied, the delay buffer is inserted into the return signal B:

$$-\alpha < Delay_{Fhalf} - Delay_{Fmax} < 0, \qquad (12)$$

$$-\alpha < Delay_D - Delay_C < 0. \qquad (13)$$

In other cases, (9) is satisfied without inserting or removing delay buffers.

IV. Timing Closure

A. Timing Constraints

Based on the observation in the previous section, we express the setup and the hold timing constraints for the flip-flops in the DVFS module and the non-DVFS module. Let $Min\_delay(FFi(ClkIn) \rightarrow FFj(DataIn))$ be the minimum delay in the set of a path whose start point is the clock-input port of a flip-flop *FFi* and whose end point is the data-input port of *FFj*. To be precise, *Min_delay* is a mapping of $\mathbf{P}^2 \rightarrow \mathbf{R}$, where **P** is the set of ports of flip-flops and **R** is the set of real numbers. The function *Max_delay* is defined, similarly. *Clock_delay*(*FFi*) denotes the clock delay between the ROOTCLK and the clock-input port of the flip-flop *FFi* shown in Fig.4.

(I)The following is the hold timing constraint when *V=VDDmax* and *F=Fmax*, for any pair (*i*, *j*) such that *FFi* is a flip-flop in the DVFS module and *FFj* is one in the non-DVFS module, or vice versa;

$$Min\_delay(FFi(ClkIn) \rightarrow FFj(DataIn)) > Clock\_delay(FFj) - Clock\_delay(FFi) + \alpha + \beta \,. \quad (15)$$

(II)The following is the setup timing constraint when *V=VDDmax* and *F=Fmax*, for any pair (*i*, *j*) such that *FFi* is a flip-flop in the DVFS module and *FFj* is one in the non-DVFS module, or vice versa;

$$Max\_delay(FFi(ClkIn) \rightarrow FFj(DataIn)) < Clock\_delay(FFj) - Clock\_delay(FFi) + (1/Fmax) - \alpha - \beta \,. \quad (16)$$

(III)Inequality (15) is the hold timing constraint when $V \neq VDDmax$ and *F=Fhalf*, for any pair (*i*, *j*) such that *FFi* is a flip-flop in the non-DVFS module and *FFj* is one in the DVFS module. When *FFi* is a flip-flop in the DVFS module and *FFj* is one in the non-DVFS module, the hold timing constraint is as follows;

$$Min\_delay(FFi(ClkIn) \rightarrow FFj(DataIn)) > Clock\_delay(FFj) - Clock\_delay(FFi) + \alpha + \beta + \gamma \,. \quad (17)$$

(IV)Inequality (16) is the setup timing constraint when $V \neq VDDmax$ and *F=Fhalf*, for any pair (*i*, *j*) such that *FFi* is a flip-flop in the DVFS module and *FFj* is one in the non-DVFS module. When *FFi* is a flip-flop in the non-DVFS module and *FFj* is one in the DVFS module, the setup timing constraint is as follows;

$$Max\_delay(FFi(ClkIn) \rightarrow FFj(DataIn)) < Clock\_delay(FFj) - Clock\_delay(FFi) + (1/Fhalf) - \alpha - \beta - \gamma \,. \quad (18)$$

Remark: When *FFi* and *FFj* are flip-flops in the same module, the timing constaints are relaxed. The inequalities by setting α = 0 to (15) and (16), are the setup and the hold timing constraints, respectively.

B. Method for Timing Closure

Next, we discuss a flow of the timing closure. The main focus is the way how to generate a circuit which satisfies the inequalities from (15) to (18) above. We use the scalable polynomial delay model (SPDM)[15] which can be generated by the curve-fitting method. By using this model, the cell delay can be expressed by the variables *c*, *s* and *v* which denote the output capacitance, the input slew and the supply voltage, respectively:

$$Delay = Delay(c,s,v) = \sum_{i,j,k=0}^{l,m,n} a_{ijk} * c^i * s^j * v^k \,. \quad (19)$$

The following is the flow for the timing closure.

Step 1: A logic circuit is synthesized and optimized under the condition of the supply voltage *V* = *VDDmax* for the DVFS module. Every path in a logic circuit is adjusted to satisfy the inequality both (15) and (16). The non linear delay model (NLDM) can be used at this point, instead of SPDM.

Step 2: The minimum supply voltage *VDDmin* is determined. As the path delay can be given like (19) by using SPDM, the functions of *Max_delay* and *Clock_delay* are expressed by polynomials of the variable *v*. So, we can obtain the minimum value *v* = *VDDmin* such that (18) is satisfied, by solving the *n*-th degree inequality.

Step 3: If (17) is satisfied for any *v* such that $VDDmin \leq v \leq VDDmax$, the timing constraints are satisfied and the timing closure is done. If not, it is necessary to add some delay buffers to fix the hold violation. To determine which and how many delay buffers are to be added, a polynomial is added to the left side of (17). It expresses the delay of an adequate delay buffer. If (17) is satisfied for any *v* such that $VDDmin \leq v \leq VDDmax$ by this addition, the timing closure can be done by adding such delay buffer. If not, it is necessary to add more delay buffers.

V. Experimental Results

We applied the clock design methodology described in Section III to the H.264/MPEG-4 LSI [2]. This LSI was implemented using our TC300C(90nm) technology[16]. Fig.5(a) shows the clock delay distribution map without using our novel clock design methodology. The horizontal axis denotes the clock delay between ROOTCLK and a flip-flop, and the vertical axis denotes the number of flip-flops whose clock delay is the value of the horizontal. The graph "DVFS(VDDmax)" indicates the clock delay distribution of the flip-flops in the DVFS-module when the supply voltage value is equal to *VDDmax*. "DVFS (VDDmin)" indicates the one when the supply voltage value is equal to *VDDmin*. The graph "non-DVFS" indicates the number of the flip-flops in the non-DVFS module. Here, we picked up flip-flops in the non-DVFS module which are interactive with flip-flops in the DVFS module, because the clock delay variation in the DVFS module gives a timing impact on only these flip-flops. The maximum clock skew is nearly 1.0ns. One remark is that the shape of the graph "DVFS(VDDmax)" and "DVFS (VDDmin)" are different. It indicates that the ratio of the clock delay under *VDDmax* and the one under *VDDmin* is different for each clock path. One clock path may become slower than another under *VDDmin*, even if the path delay of the two is same under *VDDmax*. Fig.5(b) shows the clock delay distribution map with using our novel clock design methodology. The three graphs gather around the clock delay 4.3ns, and the maximum clock skew is reduced to nearly 0.35ns.

Next, we applied our timing closure technique which is presented in Section IV. In [2], 0.90V was put in the place of *VDDmin*. We developed an SPDM cell library for the TC300C technology, and analyzed the timing of the chip. We set the supply voltage value from 1.20V to 0.70V, every 0.025V. Fig.6 shows the result of the static timing analysis. The horizontal axis shows the supply voltage, and the vertical axis denotes the timing slack of the chip. The slack for the frequency *Fhalf*(=90MHz) is shown in Fig.6. If the timing slack is positive, the timing constraints are satisfied. When the DVFS module works with the frequency *Fmax*(=180MHz), the supply voltage value is equal to 1.20V. So, the slack value of nearly 5.6ns is necessary under *VDDmax*(=1.20V). We can find that the *VDDmin* is equal to nearly 0.82V by Fig.6, as the slack value is nearly 0ns when the supply voltage value is around 0.82V.

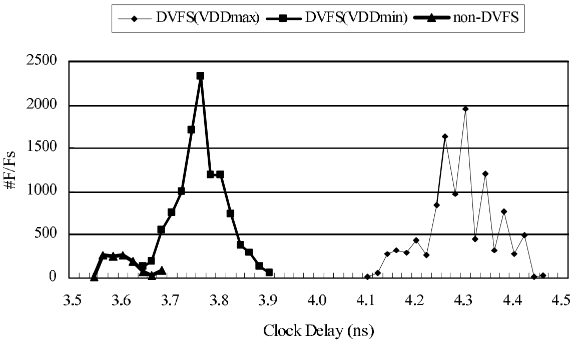

Fig.5(a). Clock delay distribution without the clock design methodology

Table 1 shows the power dissipation data of the chip. We measured the power dissipation of the LSI by two functions. One is the video data decoding and the other is the data encoding. The column "Chip-wise" denotes the power dissipation data, when the DVFS technique is employed for the whole chip. "Module-wise(0.90V)" and "Module-wise (0.82V)" denote the data, when the module-wise DVFS technique is employed for *VDDmin*=0.90V and 0.82V, respectively. The upper table shows the power dissipation data of the DVFS module. About 60% power reduction has been achieved by the module-wise DVFS, compared with the chip-wise DVFS. As for the chip-wise DVFS, it was unable to lower the voltage so much because often one of the modules in a chip requires high throughput. By setting the proposed derived *VDDmin*, about 17% power reduction has been achieved by the module-wise DVFS. The lower table shows the power dissipation data of the sum of the DVFS module and the non-DVFS modules. Nearly 9% power reduction can be achieved by the module-wise DVFS, but the effectiveness is less than what is shown in the upper table. To improve the effectiveness, it is necessary to adopt the module-wise DVFS to more than two modules.

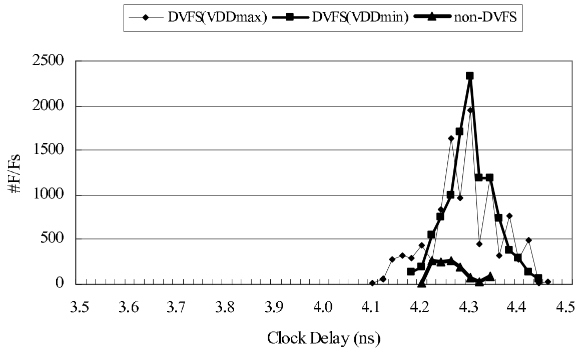

Fig.5(b). Clock delay distribution with the clock design methodology

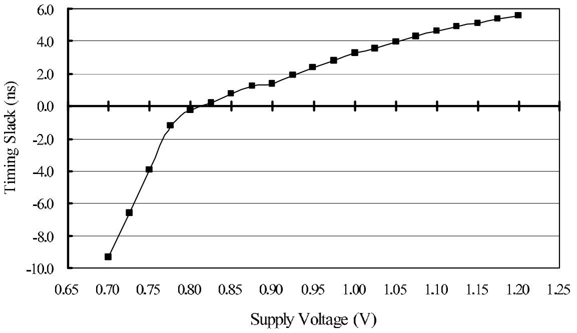

Fig.6. Relation between supply voltage and timing slack

Table 1. Power dissipation of codec LSI

DVFS module	Chip-wise	Module-wise (0.90V)	Module-wise (0.82V)
decoding	5.3mW (100.0%)	2.5mW (47.1%)	2.1mW (39.6%)
encoding	6.8mW (100.0%)	3.3mW (48.5%)	2.7mW (39.7%)

DVFS module + non-DVFS modules	Chip-wise	Module-wise (0.90V)	Module-wise (0.82V)
decoding	41.3mW (100.0%)	38.5mW (93.2%)	38.1mW (92.3%)
encoding	50.9mW (100.0%)	47.4mW (93.1%)	46.8mW (91.9%)

VI. Conclusion

We have proposed the clock design methodology for the module-wise DVFS technique. We can achieve more power reduction by the technique, as the possibility of lowering the supply voltage increases, compared with the chip-wise DVFS technique. By using DDS, instructions can be executed, even when the supply voltage is in transition. It gets rid of the performance loss. We have presented the clock design strategy to minimize the clock skew among inter-modules, even when the clock delay in each module changes, according to its supply voltage value. We have also described the condition of the timing closure which consists of the setup and the hold constraints, and shown the way of finding the minimum supply voltage value, by solving simultaneous polynomial inequalities. We have demonstrated that the 17% additional power reduction can be achieved by using the minimum supply voltage value. Adopting the module-wise DVFS technique to plural modules is a future work, in order to achieve more power reduction.

Acknowledgements

The authors would like to thank T.Yoshimori, T.Furuyama, K.Ochii, S.Imai, M.Murakata, H.Ohta, Y.Oowaki, M.Takahashi, M.Yamada, and A.Watanabe for their support. They also would like to thank T.Sakamoto and K.Miyabe for providing the SPDM environment, and thank T.Nishikawa and Y.Kikuchi for helping the static timing analysis.

References

[1]M.Fleischmann, "LongRun^{TM} Power Management – Dynamic Power Management for Crusue Processors", *http://www.transmeta.com/corporate/pressroom/whitepapers.html*, January, 2001.

[2]T.Fujiyoshi et al., "An H.264/MPEG-4 Audio/Visual Codec LSI with Module-Wise Dynamic Voltage/ Frequency Scaling", *Proc. ISSCC*, pp.132–133, 2005.

[3]S.Lee et al., "Reducing pipeline Energy Demands with Local DVS and Dynamic Retiming", *Proc. ISLPED*, pp.319–324, 2004.

[4]B.Gorjiara, N.Bagherzadeh and P.Chou, "An Efficient Voltage Scaling Algorithm for Complex SoCs with Few Number of Voltage Modes", *Proc. ISLPED*, pp.381–386, 2004.

[5]A.Chandrakasan and R.Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, 1995

[6]R.Jejurikar, C.Pereira and R.Gupta, "Leakage Aware Dynamic Voltage Scaling for Real-Time Embedded Systems", *Proc. DAC*, pp.275–280, 2004.

[7]K.Choi, W.Lee, R.Soma and M.Pedram, "Dynamic Voltage and Frequency Scaling Under a Precise Energy Model Considering Variable and Fixed Components of the System Power Dissipation", *Proc. ICCAD*, pp.29–34, 2004.

[8]A.Andrei, M.Schmitz, P.Eles, Z.Peng and B.Al-Hashimi, "Simultaneous Communication and Processor Voltage Scaling for Dynamic and Leakage Energy Reduction in Time-Constrained Systems", *Proc. ICCAD*, pp.362–369, 2004.

[9]M.Anis, S.Areibi, M.Mahmoud and M.Elmasry, "Dynamic and Leakage Power Reduction in MTCMOS Circuits Using an Automated Efficient Gate Clustering Technique", *Proc. DAC*, pp.480–485, 2002.

[10]K.Usami, N.Kawabe, M.Koizumi, K.Seta and T.Furusawa, "Automated Selective Multi-Threshold Design For Ultra-Low Standby Applications", *Proc. ISLPED*, pp.202-206, 2002.

[11]T.Kitahara, N.Kawabe, F.Minami, K.Seta and T.Furusawa, "Area-efficient Selective Multi-Threshold CMOS Design Methodology for Standby Leakage Power Reduction", *Proc. DATE*, pp.646-647, 2005.

[12]T.Kuroda et al., "A 0.9V 150MHz 10mW 4mm2 2-D Discrete Cosine Transform Core Processor with Variable-Threshold-Voltage Scheme", *IEEE JSSC*, vol.31, pp.1770 –1779, November, 1996

[13]J.Seo, T.Kim and K.Chung, "Profile-based Optimal Intra-task Voltage Scheduling for Hard Real-Time Applications", *Proc. DAC*, pp.87–92, 2004.

[14]K.Usami et al., "Design Methodology of Ultra Low-power MPEG4 Codec Core Exploiting Voltage Scaling Techniques", *Proc. DAC*, pp.483–488, 1998.

[15]*http://www.scd.magma-da.com/articles/SiliconSPDM.pdf.*

[16]*http://www.semicon.toshiba.co.jp/eng/prd/asic/doc/pdf/bce0012a.pdf.*