

# A 0.9 V Low-power 16-bit DSP Based on a Top-down Design Methodology

●Atsushi Tsuchiya ●Tetsuyoshi Shiota ●Shoichiro Kawashima

*(Manuscript received December 8, 1999)*

**A 0.9 V low-power (8.7 mW) 16-bit DSP was developed for mobile wireless use using a 0.25  $\mu\text{m}$  dual-threshold-voltage (dual-V<sub>th</sub>) CMOS process. To obtain a high-performance LSI at a low supply voltage and also to speed up the design process, we propose a new top-down design methodology in which iterations are done within the block synthesis step so that the layout can be fixed in one pass. There are four main design steps in the methodology. 1) Inter-block wires in the chip top level are routed and their precise delays are extracted from their shapes. Then, the locations of circuit blocks in the chip top level are optimized by performing timing analysis. 2) To synthesize the blocks, timing budgets are assigned according to the precise wire delays. 3) The block synthesis with the inter-block wire delays and re-assignment of the timing budgets for neighboring blocks are repeated until the timing budgets become feasible and consistent for the whole chip. 4) The entire chip layout, which involves placement and routing inside the blocks and detailed routing in the chip top level, is completed. As a result, no timing violation appears in the final timing analysis.**

## 1. Introduction

A mobile wireless terminal requires a low-power DSP to extend its battery life. When developing such a DSP, an effective way to reduce the power consumption is to decrease the supply voltage. We have developed a DSP which employs a dual multiplier and accumulator (MAC)<sup>1)</sup> and a dual-threshold-voltage (dual-V<sub>th</sub>) CMOS library<sup>2)-5)</sup> to prevent the operating frequency from falling at lower supply voltages. However, it is not easy to design a high-performance DSP within a tight schedule using only these techniques. This is because, in the conventional design methodology, many design iterations are required after the entire layout has been completed. In response to this problem, we developed a new top-down design methodology in which synthesis and re-assignment of the timing budgets are repeated so that the layout can be completed in one pass and a DSP can be developed in a short period.

In the conventional design methodology, a flat design or a bottom-up hierarchical design is performed. The flat design methodology is useful for developing a moderate-speed LSI within a short period, but it cannot be applied for a high-performance LSI because there are so many critical paths which can only be fixed with a systematic solution. In a hierarchical design methodology, on the other hand, an LSI is divided into functional blocks that are placed in the chip top level to localize the timing issue. Because it is easier to improve the performance of blocks individually than to improve the performance of the whole chip, a hierarchical design methodology, therefore, is regarded as being suitable for a high-performance LSI. However, in conventional hierarchical bottom-up design, the layouts of the blocks are completed individually then the blocks are placed in the chip top level. Then, timing analysis of the entire chip is started using the extracted

capacitances, including the capacitances of the inter-block wires. As a result, the high frequency target cannot be reached in a one-pass design process and design iterations become inevitable, because the blocks are individually designed with timing budgets that do not accurately consider the delays of inter-block wires. Therefore, in a hierarchical design methodology for a high-performance LSI, it is essential to assign appropriate timing budgets for block synthesis.

On the other hand, in recent technologies, the wire resistance per unit length is so large that long wires in the chip top level have a huge delay. Therefore, high-performance LSI design must reduce the inter-block wire length by optimizing the floorplan (i.e., the block placement and inter-block wire routing at the chip top level).

In this paper, we propose a new top-down hierarchical design methodology. We have used this methodology to develop a 16-bit DSP for mobile wireless use which is mainly composed of 13 functional circuit blocks (**Figure 1**).

With this methodology, the design conditions for blocks are properly assigned in the chip-top-level design step. To do this, we first carry out an initial floorplanning at the chip top level and obtain the precise inter-block wire delays for the floorplan obtained. Next, the floorplan at the chip

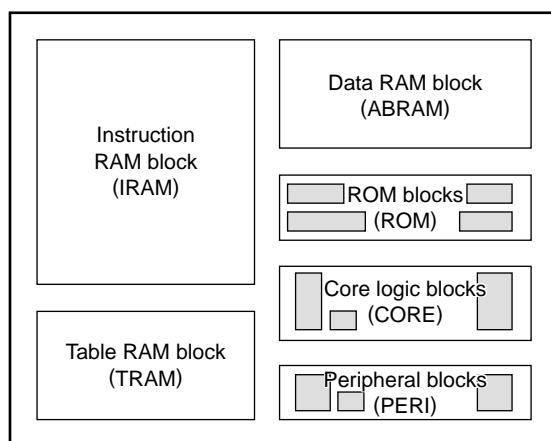


Figure 1  
13 blocks of 16-bit DSP core.

top level is optimized by using the obtained inter-block wire delays.

Then, timing budgets are assigned to each block according to the wire delays and the timing budgets are optimized by repeating synthesis and re-assignment in the block design step.

Chapter 2 of this paper explains the top-down design methodology. Chapters 3 and 4 explain the floorplan optimization and timing budgets, which are key features in the top-down design methodology. Then, Chapter 5 describes the performance of a DSP we fabricated using the top-down design methodology.

## 2. Top-down design methodology

The new top-down design methodology for developing a high-performance LSI assigns the design conditions for blocks in the chip-top-level design step. In this chapter, we explain the three key features of the new methodology.

### 2.1 Conventional bottom-up design methodology

A conventional bottom-up design methodology is shown in **Figure 2**. First, the blocks' netlists, which describe the connections between standard cells and memory macros, are synthesized from each Register Transfer Language (RTL) to satisfy the synthesis constraints for the blocks. The main purpose of these constraints is to set synthesis goals so that the blocks meet the specified timing requirements; these requirements are called the timing budgets.

Next, in the floorplanning step, the blocks, which are basically black boxes with terminals and a specified size, are placed in the chip top level, and signal, clock, and power wires are routed according to the block placement in the chip top level.

The third step performs layout, placement, and routing within each block and detailed inter-block wire routing in the chip top level. After the layout step, the timing over the whole chip is analyzed to check whether operation at the target frequency is possible. If this cannot be achieved,

the design process is repeated starting from the floorplanning or block synthesis step.

In the bottom-up design methodology, the blocks are synthesized before the floorplanning. When the new DSP was developed, the longest inter-block wire delay exceeded 2.5 ns, which is 10% of the cycle time. Therefore, synthesizing the RTLs under timing budgets which use improper inter-block wire delays would have lead to timing violations in the timing analysis; as a result, it would have been necessary to repeat the design step, starting from the block synthesis step. It would therefore have taken a long time to develop the DSP by the bottom-up design methodology.

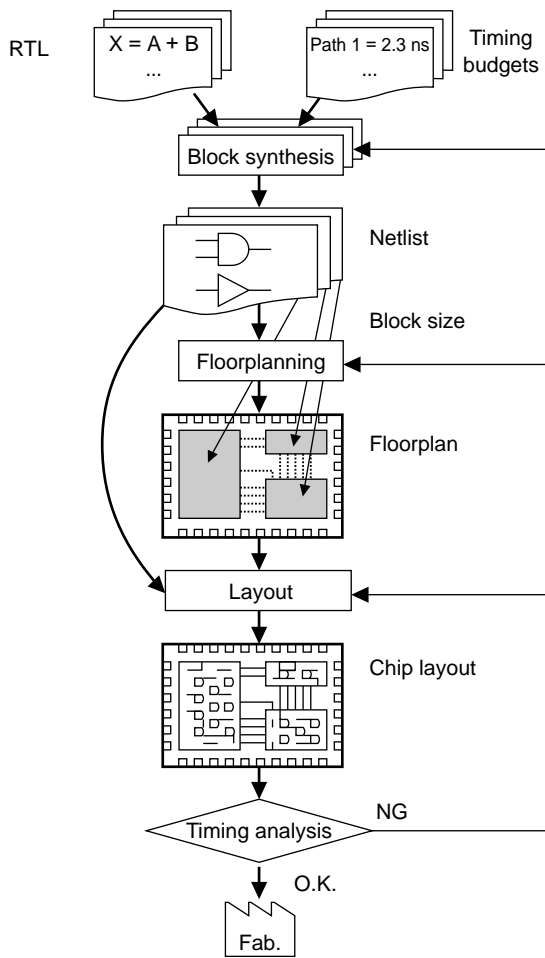


Figure 2  
Conventional design flow.

## 2.2 Proposed top-down design methodology

In this paper, we propose the new top-down design methodology shown in **Figure 3** and describe how we applied it to design the new DSP. This methodology is quite different from the conventional one, because we carry out the floorplanning step before the block synthesis step in order to assign accurate timing budgets.

Accurate timing budgets are obtained as follows. First, the RTL design of blocks is pre-synthesized with initial timing budgets; the obtained pre-synthesized netlists are mainly used to predict the block sizes. Second, floorplanning is carried out using the predicted block sizes. A timing analysis optimizes the floorplan to further improve the DSP's performance in this step. This

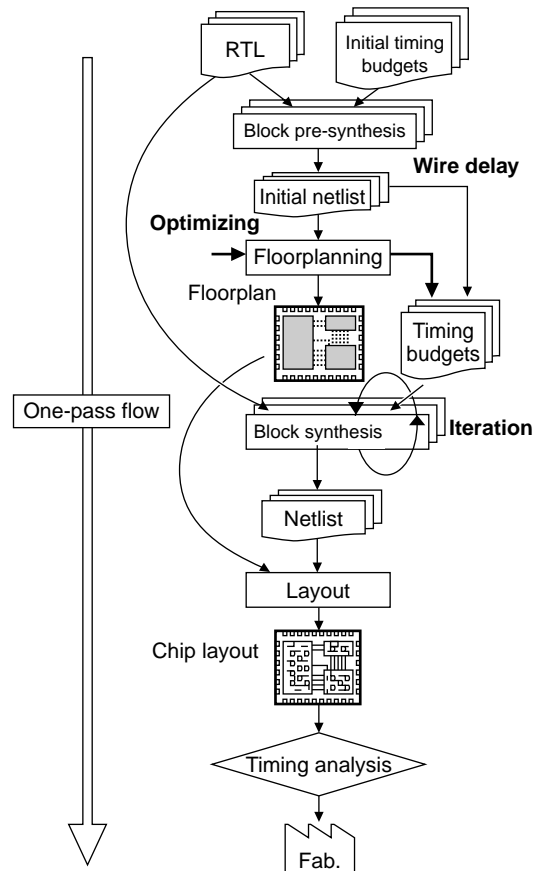


Figure 3  
New top-down design flow.

analysis takes the extracted inter-block wire delays into account, which is described in detailed in Chapter 3. Third, we synthesize blocks from their RTLs with timing budgets that take the inter-block wire delays into account. Fourth, we introduce a precision improvement method for timing budgets which repeats between block synthesis and timing budget re-assignment. This is described in Chapter 4.

After the floorplanning, layout is performed and the timing is analyzed for the entire chip. When the timing budgets are properly assigned and achieved in the block synthesis and layout, no timing violation appears after the chip layout is completed. Thus, a high-performance DSP can be quickly designed in a one-pass layout with no need to change the design.

### 3. Timing analysis driven floorplan optimization

We propose the floorplan optimization shown in **Figure 4**. First, we do a block pre-synthesis to

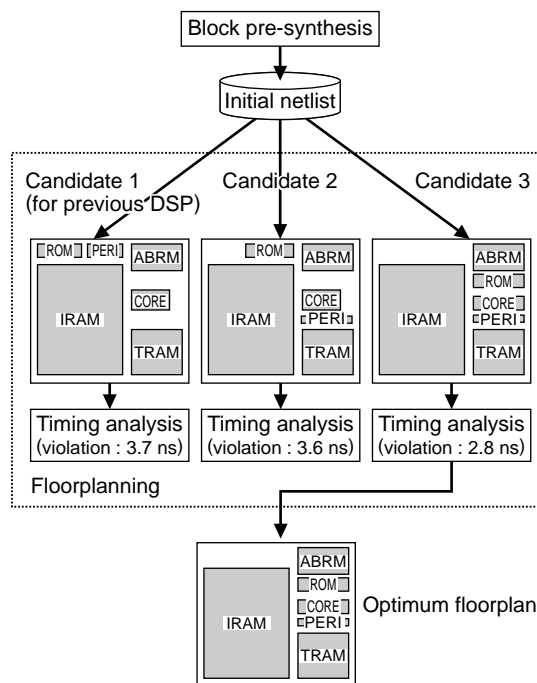


Figure 4 Floorplan optimization.

obtain the netlists for estimating the block sizes. This synthesis uses rough initial timing budgets which are derived from the previous DSP design. Then, we make a floorplan. The designer optimizes the floorplan to reduce the inter-block wire delays and to improve the delays on critical timing paths. To evaluate the floorplans of the DSP in the optimizing step, we performed a timing analysis using the initial netlists and the load of the inter-block wires. The method for obtaining and using accurate inter-block wire delays from the extracted wire resistances and capacitances is explained in Chapter 4.

The floorplan of the DSP was developed by improving on Candidate 1 (Figure 4), which is the floorplan of the previous 2.5 V-operation DSP with a 0.25  $\mu\text{m}$  standard CMOS technology. The improvement was done by analyzing the timing using the inter-block wire delays with the 0.25  $\mu\text{m}$  dual Vth CMOS technology and a 1.0 V operating condition. Timing analysis with a 20 ns cycle time showed that Candidate 3 in the figure had the smallest timing violation of the three candidates we tried. Therefore, Candidate 3 is the optimum one for this technology. Figure 4 shows that floorplan optimization reduces the timing violation by 0.9 ns, which corresponds to a 5% increase in the operating frequency.

### 4. Timing budget considering inter-block wire delays in floorplan

Applying a hierarchical design to a high-performance LSI requires assignment of the correct timing budgets. These should be calculated in consideration of the inter-block wire delays extracted from the floorplan. We also propose a method for improving the precision of timing budgets in which multiple iterations of synthesis and timing budget re-assignment are performed.

#### 4.1 Timing budget equation

The timing budget definition is shown in **Figure 5**. In this figure, block A has already been synthesized and block B is the synthesis target

block. The timing budget for block B is given for the path from an input pin to a flip-flop (FF) in block B. The delay from an FF in block A to an output pin in block A, which is called the “external block delay,” can be calculated using the synthesized netlist. In the design of a synchronous LSI, the delay between FFs must be less than the machine cycle time when the clock skew is negligibly small. Hence, the timing budget is defined by the following equation.

$$\text{Timing budget} = \text{Cycle time} - \text{External block delay} - \text{Wire delay} \quad (1)$$

Assigning correct timing budgets involves estimating the correct external block delays and wire delays. In the timing analysis, the wire delay calculation uses the values of wire capacitance and resistance extracted from the layout data. The method for interpreting the precise wire delays is explained in Sections 4.2 to 4.3. The method for alternately improving the precisions of the timing budgets and external block delays is explained in Section 4.4.

#### 4.2 Wire load estimation based on floorplan

In the technology we used for the new DSP, the delay at synthesis is calculated using the delay table for each standard cell and the wire load

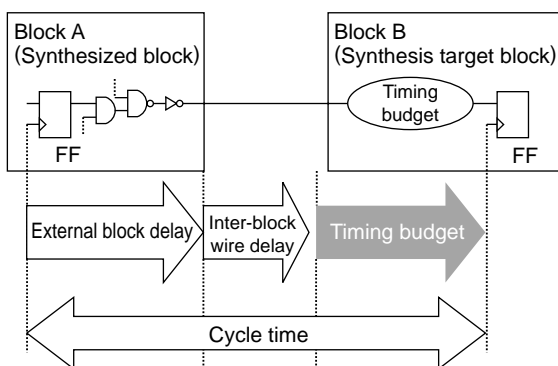


Figure 5  
Timing budget for synthesis target block.

connected to the pins. Therefore, an accurate delay for the synthesis can be derived only by modifying the wire load.

A wire load consists of a wire capacitance and resistance. In a conventional design methodology, the capacitances of wires inside the blocks and in the chip top level are estimated from the number of fan-outs and the areas of the blocks. The capacitance table used for this estimation is called the Wire Load Model (WLM). The WLM is useful for wires inside blocks because the block area is not so large and therefore these wires are short. In this case, there are no serious differences between a capacitance obtained from WLM calculation and one extracted from the layout; therefore, there are no violations inside any of the blocks in the timing analysis step after the layout has been completed.

However, the lengths of the inter-block wires in the chip top level of the new DSP differ widely. **Figure 6** shows the distribution of inter-block wire lengths in the chip top level of the new DSP. The wire lengths range from 0.12 to 6.4 mm. If we estimate the capacitances of inter-block wires in the chip top level with WLM, wires that have the same fan-out number and are on the same hierarchy will be estimated to have the same capacitance. The estimated capacitances of these wires, therefore, will not reflect their real length.

In the case of the top-down design methodol-

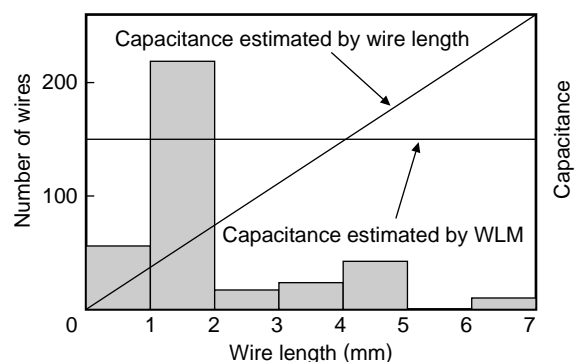


Figure 6  
Lengths of inter-block wires and capacitance estimation.

ogy (Figure 3), the floorplan step is carried out before the block synthesis step and the wire lengths can be extracted from the floorplan. The extracted wire lengths can then be used to estimate the wire capacitances more accurately than can be done using WLM. Then, the wire delays are calculated from the estimated capacitances and the delay tables of the cells driving the wires. We also include the effect of wire resistance in the wire delay calculation; this is described in Section 4.3.

### 4.3 Wire capacitance correction considering wire resistance

The wires have a resistance as well as a capacitance. This resistance affects the wire delay when a wire is long. In Figure 7 (a),  $T_{real}$  shows the simulated delay as a function of length for a wire driven by an inverter cell from the 0.25  $\mu\text{m}$  dual-Vth CMOS library with a medium driving

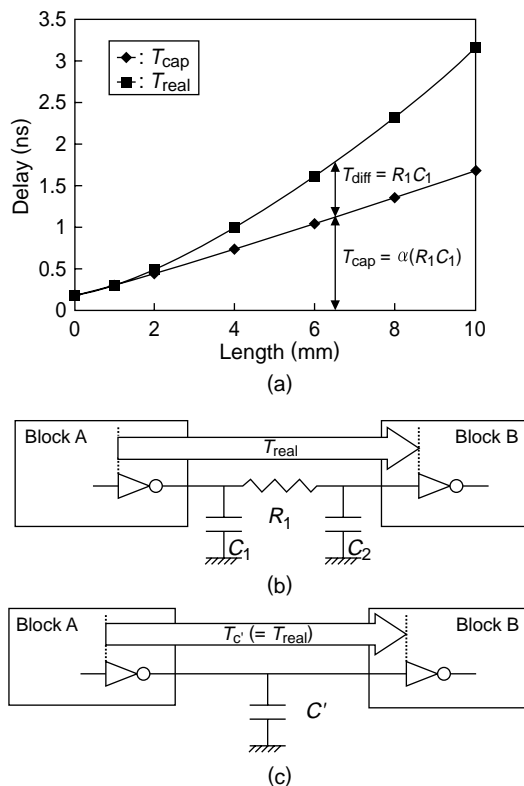


Figure 7 Wire delay calculation.

capacity.  $T_{real}$  shows the average of the rise and fall propagation delays. The wire is assumed to be a first metal line of the minimum width and is represented by a simple  $\pi$  model (Figure 7 (b)) that consists of resistor  $R_1$  and capacitors  $C_1$  and  $C_2$  that are extracted from the floorplan.<sup>6)</sup>  $T_{real}$  is calculated from these elements.

However, as shown in Figure 7 (c), the delay table used in the block synthesis step only takes the capacitance into account. In Figure 7 (a),  $T_{cap}$  shows the delay model used in the block synthesis. The difference between  $T_{real}$  and  $T_{cap}$  is referred to as  $T_{diff}$  and is equal to the product of  $R_1$  and  $C_2$ . Since the intrinsic delay of the inverter cell is very small compared with other delays,  $T_{cap}$  is nearly proportional to  $C_1 + C_2$ . The proportional coefficient is shown as  $\alpha$  in Figure 7 (a). The optimum value of  $\alpha$  is about 800  $\Omega$  for this library. The delay calculated in synthesis using  $C_1 + C_2$  is smaller than  $T_{real}$  because  $T_{diff}$  is not considered.

We propose a capacitance correction so that the wire resistance is taken into account. In the delay table, the load is treated as the purely capacitance element  $C'$  in Figure 7 (c). When  $T_{c'}$  is defined as the delay caused by  $C'$ ,  $C'$  must be larger than  $C_1 + C_2$  in order to make  $T_{c'} = T_{real}$ . Because the capacitance corresponding to  $T_{diff}$  is  $(T_{diff}/\alpha)$ ,  $C'$  is obtained as follows:

$$C' = (C_1 + C_2) + (T_{diff}/\alpha) \quad (2)$$

When using the value  $C'$  in the synthesis, the wire delay is estimated correctly. In practice, extracted values  $C_1$ ,  $C_2$ , and  $R_1$  are calculated to get  $T_{cap}$  and  $T_{diff}$ . Then, using Equation (2),  $C'$  is obtained and used for the synthesis.

### 4.4 Alternate timing budget/external block delay precision improvement

The external block delay, which is another element used to calculate the timing budget, depends on the netlists of the neighboring blocks. When the netlist of one block is synthesized, using the new external block delay, the timing bud-

gets of the neighboring blocks should be adjusted to improve their precision. However, the external block delays depend on the timing budgets of other external blocks and we cannot decide the timing budgets easily.

We propose a method in which the timing budgets are precisely calculated by alternately repeating block synthesis and timing budget re-assignment. **Figure 8** shows how this method gradually improves the precision of the timing budgets between two blocks. The variables in step 1 are an inter-block wire delay from a floorplan, two external block delays from the initial netlists for blocks A and B, and a cycle time that must be satisfied. A new timing budget (TB\_B1) for block B based on the external block delay of A0 and the wire delay in step 2 is assigned. Then, a new netlist of block B is synthesized with the new timing budget and a new external block delay, B1, is derived. In step 3, block A is assigned a new tim-

ing budget (TB\_A1). This repetition between block synthesis and timing budget re-assignment continues until the total delay time between FFs equals the cycle time. When the timing requirement is satisfied as shown in step X, the timing budgets and netlists at that instant satisfy the timing requirement for the whole chip.

### 5. Performance of fabricated DSP

Using the new top-down design methodology, we developed a 16-bit DSP with a 0.25 μm dual-Vth CMOS process. **Figure 9** shows its photograph, and **Table 1** lists its main features. For the logic blocks, a multi-threshold CMOS (MT-CMOS) technology<sup>2),3)</sup> reduces the standby power consumption. SRAM and ROM macros<sup>4),5)</sup> have been developed for operation at a supply voltage of 1 V.

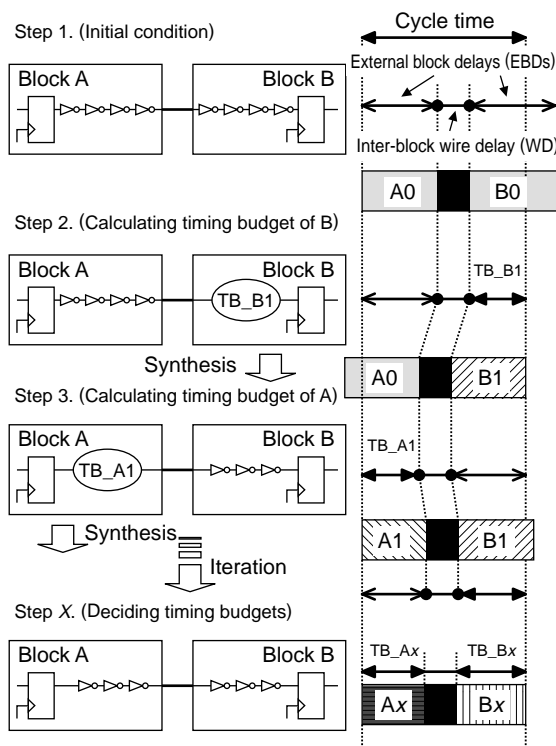


Figure 8 Precision improvement for timing budgets and external block delays.

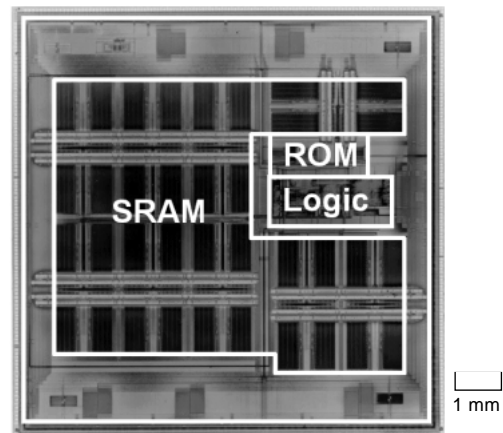


Figure 9 Photograph of 16-bit DSP chip.

Table 1 Main features of 16-bit DSP.

Technology	0.25 μm, 3-metal, Dual-Vth CMOS
pMOS Vt	-0.45 V/-0.20 V
nMOS Vt	0.45 V/0.20 V
Die size	12.0 mm × 12.0 mm
Number of transistors	330 kTr + 32 Kb-RAM × 36 + 32 Kb-ROM × 3
Supply voltage	0.9 V (core) 2.5 V, 3.3 V (I/O)

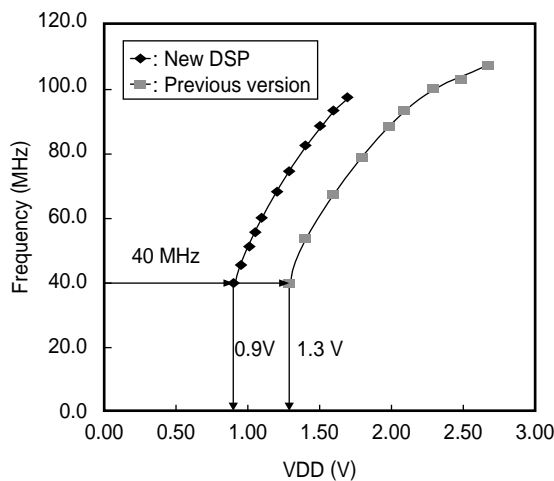


Figure 10  
Operating frequency versus VDD: half-rate CODEC.

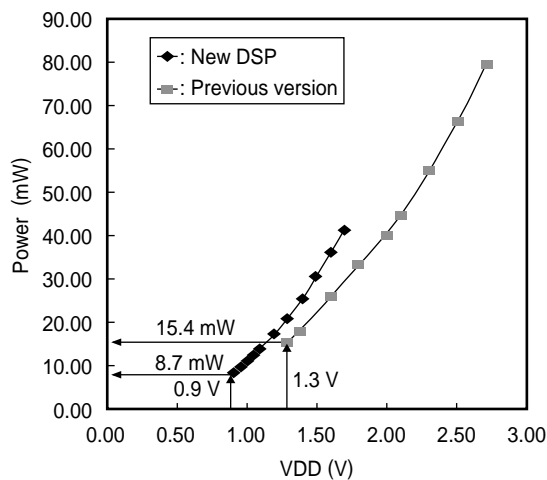


Figure 11  
Power consumption versus VDD: half-rate CODEC.

Figures 10 and 11 show the operating frequency and power consumption of the new DSP and the previous version when they perform half-rate compression/decompression (CODEC) at various values of VDD. Both DSPs originated from the same RTLs, but the previous version was developed with a 0.25  $\mu\text{m}$  standard Vth CMOS process with a conventional design flow. As Figure 10 shows, an operating frequency of 40 MHz, which is sufficient for Japanese PDC half-rate CODEC, is realized at 0.9 V in the new DSP and

at 1.3 V in the previous version. Figure 11 shows that the power consumption of the new DSP at 0.9 V is 8.7 mW, which is 40% less than the 1.3 V value of 15.4 mW for the previous version.

## 6. Conclusion

A top-down design methodology was proposed for high-performance LSI development. A precise timing analysis at the chip top level is a key feature of this methodology. Floorplans are optimized using this timing analysis. Timing budgets reflect accurate inter-block wire delays. To calculate the inter-block wire delays precisely in the synthesis step, the capacitance values obtained from the floorplan provide delays that include the effect of wire resistance. After multiple synthesis iterations, the timing budgets become reasonably accurate and consistent and a netlist showing a high-performance is obtained. Using the top-down design methodology, we developed a 16-bit DSP with a 0.25  $\mu\text{m}$  dual-Vth CMOS process. Even at a 0.9 V supply voltage, the operating frequency of this DSP exceeds 40 MHz. The power consumption at 0.9 V is 8.7 mW, which is 40% less than the 1.3 V value of 15.4 mW for the previous version.

## References

- 1) Ishihara et al.: Low Power Consumption Digital Signal Processor: Hi-Perion, *FUJITSU Sci. Tech. J.*, **36**, 1, pp.56-62 (June 2000).
- 2) I. Fukushi, R. Sasagawa, and W. Shibamoto: Dual-Vth 0.25  $\mu\text{m}$  CMOS Cells and Macros for 1 V Low-power LSIs, *FUJITSU Sci. Tech. J.*, **36**, 1, pp.72-81 (June 2000).
- 3) T. Shiota et al.: A 1 V, 10.4 mW Low Power DSP core for Mobile Wireless Use, *VLSI Symp. VLSI circuits Dig. Tech.*, p.13 (June 1999).
- 4) I. Fukushi et al.: A Low-Power SRAM Using Improved Charge Transfer Sense Amplifiers and a Dual-Vth CMOS Circuit Scheme, *VLSI Symp. VLSI circuits Dig. Tech.*, p.142 (June 1998).
- 5) R. Sasagawa et al.: High-speed Cascode

Sensing Scheme for 1.0 V Contact-programming Mask ROM, VLSI Symp. VLSI circuits Dig. Tech., p.95 (June 1999).

6) P. Penfield and J. Rubinstein: Signal delay in

RC tree networks. *IEEE Trans. Computer-Aided Design*, **CAD-2**, pp.202-211 (July 1983).



**Atsushi Tsuchiya** received the B.S. and M.S. degrees in Information Engineering from Tohoku University, Sendai, Japan in 1996 and 1998, respectively. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1998, where he has been engaged in research and development of high-performance LSIs.

E-mail: atsuchiya@flab.fujitsu.co.jp



**Shoichiro Kawashima** received the B.S. degree in Physical Engineering from the University of Tokyo, Tokyo, Japan in 1982. He joined Fujitsu Ltd., Kawasaki, Japan in 1982, where he was engaged in the development of 16 Kb to 16 Mb MOS static RAMs. Since 1994, he has been with Fujitsu Laboratories Ltd., Kawasaki, Japan researching low-power SRAMs and DSPs. He is a member of the IEEE, the Japan Society of Applied Physics (JSAP), and the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

E-mail: poosan@flab.fujitsu.co.jp



**Tetsuyoshi Shiota** received the B.S. and M.S. degrees in Electronic Engineering from Nagoya University, Nagoya, Japan in 1988 and 1990, respectively. He joined Fujitsu Laboratories Ltd., Atsugi, Japan in 1990 and received a Ph. D. from Nagoya University in 1997. He is currently engaged in research of low-power CMOS circuits. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

E-mail: shiota@flab.fujitsu.co.jp