Design of Asynchronous Library and Implementation of Interface for Heterogeneous System

Hwi-Sung Jung, Joon-Il Lee and Moon-Key Lee

Dept. of Electronic Engineering, Yonsei University, Seoul 120-749, Korea

Abstract

We designed asynchronous event logic library with 0.25um CMOS technology and interface chip for heterogeneous system with high-speed asynchronous FIFO operating at 1.6GHz. Optimized asynchronous standard cell layouts and Verilog models are designed for top-down design methodology. A method for mitigating a design bottleneck when it comes to tolerate clock skew is described. This communication scheme using clock control circuits, which is used for the free of synchronization failures, is analyzed and implemented. With clock control circuit and FIFO, high-speed communication between synchronous modules operating at different clock frequencies or with asynchronous modules is performed. The core size of implemented high-speed 32bit-interface chip for heterogeneous system is about 1.1mm x 1.1mm.

I. INTRODUCTION

As microprocessor performance moves into the GHz speed, the high-speed asynchronous design is becoming challenge due to the disadvantageous power and speed aspects in synchronous designs. Asynchronous circuit design techniques in both system level and circuit level attract a lot of attentions in recent years. System timing in asynchronous circuit, which is absence of global clock, is performed by the elements themselves [1],[5]. The next-generation on-chip systems will consist of multiple independently synchronous modules and asynchronous modules for higher performance. To perform high-speed communication between synchronous modules operating at different clock frequencies or with asynchronous modules, a reliable communication scheme is needed. A novel communication scheme using PCC (Pausible Clocking Control) circuits is one method to communicate with synchronous modules via asynchronous FIFO (First-In First-Out) communication channels [2]. Fig.1 shows this communication scheme with handshake.protocol [3].

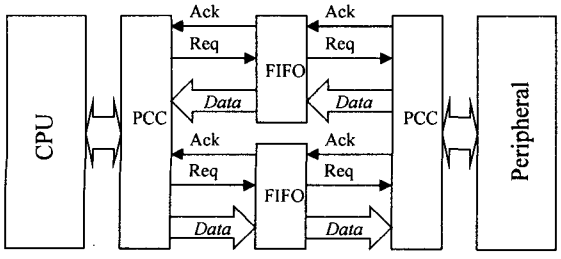

Fig.1. Communication scheme with PCC and FIFO.

It is unnecessary to consider the communication delay between blocks using asynchronous handshaking.

In order to develop compatibility and usability of circuit design, we designed asynchronous event logic library that is useful to perform handshake protocol effectively. This library includes physical layout elements and verilog timing models for top-down design methodology. Using this library, we implement high-speed asynchronous FIFO and interface chip on ASIC 0.25um CMOS chip. This FIFO operates at 1.8GHz and two separated modules can communicate with each other via FIFO.

The rest of this paper is organized as follows. Section 2 reviews asynchronous library, key components used in asynchronous design. Section 3 describes the design and implementation of FIFO and interface circuit unit for heterogeneous system. Section 4 describes the simulation results. Section 5 concludes the paper with some remarks on the future system design.

II. ASYNCHRONOUS LOGIC LIBRARY

Asynchronous logic library that is used in AMULET [4] is generated with 0.25um CMOS process. This library includes optimized cell layout with 8.8um-height and verilog timing model for the back-end and front-end design, respectively. The cell height is calculated as 10 tracks multiplied 10 by 0.88um that is one route pitch. Our implementation of library cell layout shows the equally same height for all logic cells and it enables to place and route effectively for building large blocks. To predict the behavior of the design, the library cells should be

characterized for performance. Characterization enables designers to move from the transistor level to the logic level. The gate in Fig.2 is characterized in terms of the input slew rate *Tin* and output capacitive load, C_L. Timing

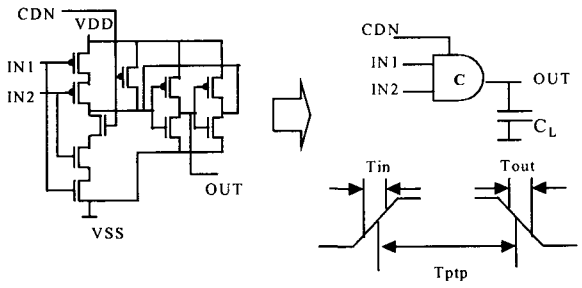

Fig.2. Delay characterization for Muller-C gate.

Information such as timing arcs (delays from an input pin to an output pin), slew rates on output pins, timing checks such as setup and hold, tristate delays, etc. are modeled as part of delay characterization [6]. Both the input to output pin to pin (ptp) delay, *Tptp*, and the output slew rate, *Tout*, are required for gate level synthesis and delay calculation tools. *Tin* and *Tout* are measured between two pre-determined edge threshold values and *Tptp* is measured between pre-determined delay threshold values [8],[9]. The gate is simulated with HSPICE (0.25um BSIM3V3 model) to obtain delay models for *Tout* and *Tptp*.

Fig.3 shows the library characterization data flow and layout verification flow. With the scale-down spacing such as 0.25um CMOS process, the p-n-p-n path in the internal cores of CMOS IC's is further sensitive to latchup [6],[7]. The methodology to verify the layout having latchup/ESD

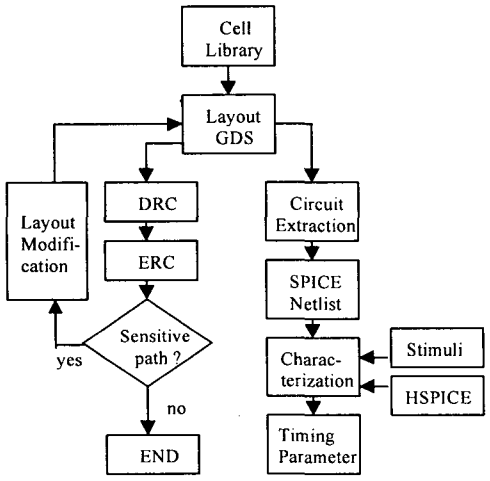

Fig.3. Layout verification flow and library characterization data flow.

sensitive paths is shown in Fig.3, where the DRACULA™ is used to check the spacing of the sensitive paths. By using the DRC and ERC functions of the DRACLA™, the latchup/ESD sensitive paths can be found and replaced by the way of insensitive layout styles.

Ring oscillator simulation and design rules determine asynchronous library specification. This ring oscillator simulation decides optimized transistor size that is used to minimize the delay, power and power-delay product. Fig.4 shows structure of ring oscillator with 21 stages. The isolation buffer in Fig. 4 is used to decrease the effects of transition time caused by large load from divider block.

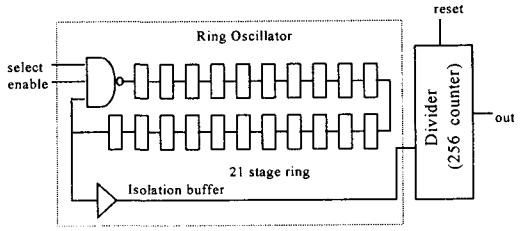

Fig.4. Structure of ring oscillator.

In order to optimize the low power and high-speed, we used following methods. The first method for low power is like this: Ring oscillator simulation shows that as the size of transistor getting smaller, the power consumption also getting down because the parasitic capacitance is getting smaller. But, our suggestion is to optimize power consumption with considering logic threshold voltage of gate. This optimization results shows that the size and power of transistor are larger than those of minimized gate. The second method for high-speed is like this: Recursive SPICE simulation determines the optimized transistor size of PMOS and NMOS in order to minimize gate delay. This gate delay is calculated as average value from rise delay and fall delay by simulation that three gates are connected in series with standard load located behind each gate. This standard load is four inverters and routing capacitance. Table 1 shows the optimized layout area and gate delay for some asynchronous library.

	Cell Size	Rise Prop.	Fall Prop.	Rise (Tr).	Fall (Tf).
Muller-C	8.8 x 8.8um²	0.142ns	0.110ns	0.122ns	0.083ns
Trans.-latch	10.56 x 8.8um²	0.130ns	0.203ns	0.201ns	0.095ns
Mutex	15.84 x 8.8um²	0.130ns	0.216ns	0.152ns	0.009ns
Toggle	29.92 x 8.8um²	0.143ns	0.216ns	0.196ns	0.109ns
Select	47.52 x 8.8um²	0.235ns	0.248ns	0.176ns	0.114ns

Table 1. Layout area and gate delay for asynchronous cells.

With this same 8.8um-height of layout, the physical library can be ported into the automatic place & route tool such as Silicon Ensemble™ and Apollo™ software to generate optimized layout effectively. The followings are the characteristics of implemented physical library designed with 0.25um CMOS process.

- This library is intended to function with a 2.5V.
- This library is to be designed to support flipping and abutting cell rows.
- We have normalized on a transistor width P/N ratio for this library of 1.3:1.
- The horizontal track 1 and track 10 are coincident with the cell AB (Abutment Box) bottom and top, respectively.
- It is preferred to place all input and output ports along one "Central Horizontal Grid" for all cells.
- All input and output ports are located on the valid position.
- Tap connections placed at the cell ends (left and/or right) are best placed directly beneath their power supply bus metals.

III. SYSTEM DESIGN

In order to perform communication in heterogeneous system such as Fig.1, asynchronous high-speed FIFO channel is needed for burst transmission. We implemented 4-stages micropipeline [4] FIFO that is self-timed circuits using two-phase or four-phase communication protocol and a bundled data format. Micropipeline falls into one of two types as follows: Transparent latch style and capture-pass latch style [4]. A block diagram of the transparent latch micropipeline FIFO is shown in Fig.5. Using asynchronous logic library, two types of micropipeline FIFO are designed with optimized method and are compared in aspects of layout area and delay time. The use of transparent latch in transparent latch micropipeline stage greatly simplifies the required control circuit than the control circuit of capture-pass micropipeline stage.

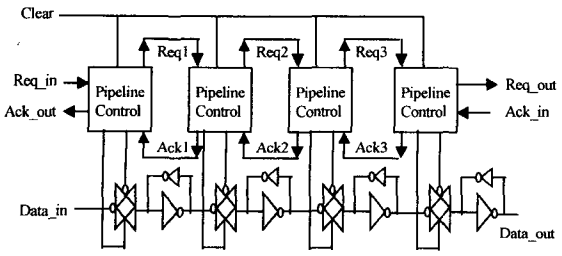

Fig.5. Transparent latch 4-stages micropipeline FIFO.

Table 2 shows the layout area of each micropipeline stage and the simulated delay, again with appropriate output loading and full loading capacitance for a datapath latch. HSPICE (BSIM3V3 model with Nominal-Nominal strength) has been performed on extracted layout from the implemented design for the nominal case (Vdd=2.5V, at 25□C temperature). The transparent latch micropipeline is performed faster than capture-pass micropipeline. Measured control signals and data in-out signals on micropipeline stage are shown in the waveforms of Fig.6. The valid data is hold when *Req* is high, then as soon as *Ack* is rising the data is transmitted to the latch output when *Req* is low. These results show that once valid data is presented at the latch input this data will be propagated to the latch output in 0.62ns for transparent latch FIFO operating at 1.6GHz.

	Transparent latch FIFO	Capture-Pass FIFO
Control circuit	40.9 x 17.2 um^2	35.3 x 17.2 um^2
Datapath latch	12.3 x 9.2 um^2	21.1 x 19.4 um^2
Req_in to Req_out	0.73ns	0.42ns
Data_in to Data_out	0.62ns	0.89ns

Table 2. Layout area and simulated delay for each FIFO. (0.25um four-layer metal CMOS process)

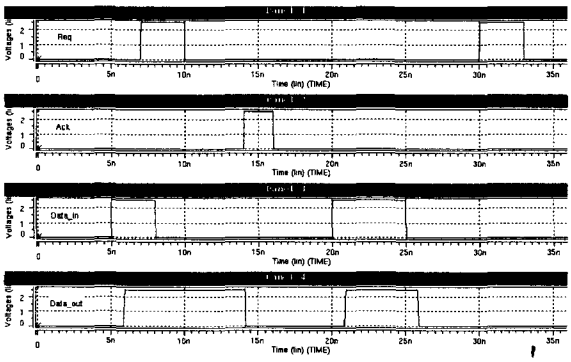

Fig.6. SPICE result on transparent latch FIFO.

Another key element for heterogeneous system except FIFO is interface circuit to transfer data between two modules that have different clock frequency and self-timed circuit. We used the pausible clocking scheme [1], which the local clock is paused or stretched to ensure that the handshaking signal satisfies setup and hold time constraints with respect to the local clock, to implement interface circuit. In this scheme, communication between a module and the FIFO is done using request/acknowledge

handshake protocol. This method adjusts individual synchronous module's local clock, when necessary, to avoid synchronization failure. Using this scheme, the synchronous module communicating with asynchronous peripheral latches the handshaking signals from the asynchronous module by stopping or stretching its own clock. We implemented this pausible clocking scheme as PCC (Pausible Clocking Control) [1] circuit style. Fig.7 shows the block diagram of proposed scheme. In the block

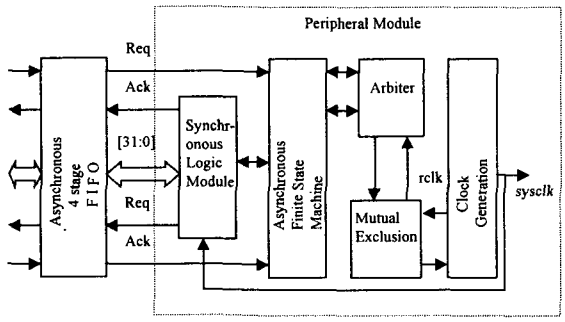

Fig.7. Block diagram for interface using FIFO and PCC.

the clock generator should be designed as ring oscillator methods or PLL (Phase Locked Loop) methods. We used ring oscillator to produce the system clock. SOC (System On a Chip) should be designed with reusable component such as IP (Intellectual Property) block. This IP block which is standardized should be reused with little or no modification, because this IP is highly optimized for performance and speed. Ideally, this interface circuit should replace a portion of the system clock generation unit in some IP blocks to transfer data between another IP that has different clock frequency.

IV. EXPERIMENTAL RESULTS

We constructed 32-bit interface chip for heterogeneous system as shown in Fig.8. First, the asynchronous logic library, which includes layout cells and verilog models, is generated and characterized for optimized layout and timing performance. Second, we implemented proposed system using HDL with optimized timing model for functional verification and performed extensive SPICE simulation on extracted layout from the design done by Virtuoso™ layout editor. We used 0.25um CMOS process design rule with four-layer metal. With this interface chip 32bit data transfer is possible between two modules which clock frequency is different. The size of the core is about 1.1mm x 1.1mm.

The timing trace in Fig.9 shows a simulation result including handshake and data signals. This result clearly indicates that the clocks do become stretched. The first event on *Req* (a rising transition) is acknowledged with pausing *sysclk*, and then second event (a falling transition) causes *sysclk* to be paused for about 0.2ns. As a result, this module operates at 2.2GHz, sysclk. With the addition or subtraction of ring oscillator stages, clock frequency could be changed to higher or lower. Because the module operates at higher clock frequency than the FIFO, the FIFO never occur bottleneck between them.

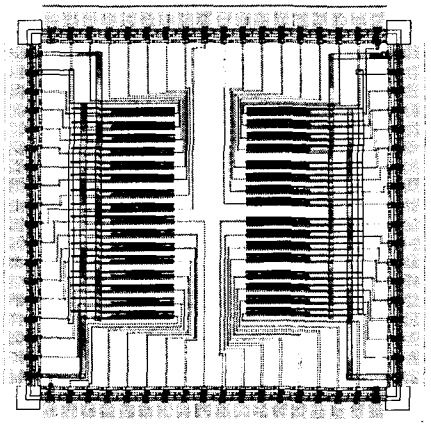

Fig.8. The chip layout.

V. CONCLUSION

We implemented effectively optimized asynchronous logic library by characterization. Using this cell library, we designed high-speed asynchronous micropipeline FIFO operating at 1.6GHz for heterogeneous system channels. New communication scheme, which is based on the pausible clocking scheme, is implemented with FIFO and PCC circuit on 0.25um CMOS chip. The resulting system functions to the local clock frequency of 2.2GHz that is limited by the ring oscillator.

In the future, we plan to implement the heterogeneous system using 32bit RISC processor and adjust at 3D graphic accelerator chip. In 3D graphic chip, there is an asynchronous block that is used to transfer data between processing block and rendering block. In addition, we will investigate a new clocking control circuit for high-speed and low power for mobile equipment.

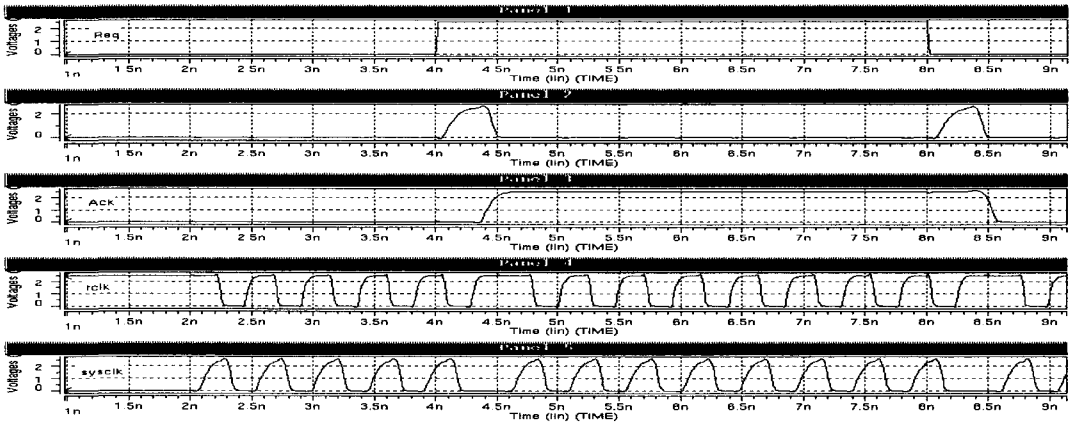

Fig.9. SPICE result on pausible clocking control circuit system.

ACKNOWLEDGEMENT

This work was supported by a grant from the Korea Research Foundation.

REFERENCES

[1] K.Y. Yun & A.E. Dooply, "Pausible Clocking-Based Heterogeneous Systems", , *IEEE Transactions on VLSI System*, Vol.7, No.4, Dec. 1999.

[2] M.R.Greenstreet "Implementing a STARI chip", *in Proc. Int. Conf. Computer Design*, pp.38-43, 1995.

[3] Furber, S.B., & Day, P., "Four-Phase Micropipeline Latch Control Circuits", *IEEE Transactions on VLSI System*, Vol.4, No.2, June 1996.

[4] N.C.Paver. "The design and implementation of an asynchronous microprocessor", *Ph.D thesis, University of Manchester*, 1994.

[5] Shyh-Jye and I-Yao Chuang. "Low-power Globally Asynchronous Locally Synchronous Design Using Self-Timed Circuit Technology." *IEEE International Symposium on Circuits and Systems*, pp.1808-1811, June 1994.

[6] Binay Ackalloor, Dinesh Gaitonde, "An Overview of Library Characterization in Semi-Custom Design", *IEEE custom integrated circuits conference*, pp.305 – 312, May 1998.

[7] Ming-dou Ker and Jeng-Jie Peng, " Layout Design and Verification for Cell library to improve ESD/latchup Reliability in Deep-Submicron CMOS Technology", *IEEE custom integrated circuits conference*, pp.537 – 540, May 1998

[8] Dake Liu and Christer Svensson, "Trading Speed for Low Power by Choice of Supply and Threshold Voltages", *IEEE journal of solid-state circuits*, Vol.28, No.1. pp.10-17, Jan.,1993.

[9] Richard X.Gu and Mohamed I.Elmasry, "Power Dissipation Analysis and Optimization of Deep Submicron CMOS Digital Circuits", *IEEE journal of solid-state circuits*, Vol.31, No.5, May 1996.