ASYNCHRONOUS CONTROL OF LOW-POWER GATED-CLOCK FINITE-STATE MACHINES

Bengt Oelmann and Mattias O'Nils

Department of Information Technology, Mid-Sweden University
S-851 70 Sundsvall, Sweden
{Bengt.Oelmann@ite.mh.se}

ABSTRACT

An efficient approach to reduce power consumption in a synchronous Finite-State Machine (FSM) is to de-compose it, according to a partitioning algorithm, to a number of sub-FSMs that interact through some communication signals. Only one sub-FSM is clocked at a time and low power operation is obtained by only clocking the active sub-FSM. In this paper we introduce a new asynchronous communication control for the interacting sub-FSMs, which reduces the total capacitance switched by the system clock. Experimental results show that this leads to significant power savings when the FSM is partitioned into many sub-FSMs.

1. INTRODUCTION

The dominating source of power dissipation in digital CMOS Integrated Circuits (IC) originates from *dynamic* power dissipation [1]. There are several methods that can be used for minimizing the dynamic power dissipation; (1) Reducing the power supply voltage is an efficient technique since $P \sim V^2$. This will increase the circuit delay but it can be compensated by increasing the parallelity [2]. (2) Reduction of parasitic capacitances can be made on different levels of abstractions, from optimization on architectural level down to IC technology optimization. (3) Reduction of switching activity can be made through logic optimization and by powering down parts of the circuit that are not active by gating the clock signal.

Gating the clock for larger parts of a design can be made by sensing certain status bits that control the power-down mode. For data paths it is possible to power down the whole circuit or parts of it by pre-computation of a clock-enable signal on the basis of the input data values [3].

In this work we present an asynchronous communication control circuit that is to be used for a power-down scheme for FSMs. In the next section we review the related work. Then the proposed asynchronous clock control is presented. Finally we conclude with experimental results.

2. RELATED WORK

In recent years, the most common approach to low-power FSM design has been to divide the FSM into a number of smaller sub-FSMs where only one of these is active at a time. The total switching capacitance can thereby be reduced. Dasgupta et al. [4] present a synthesis method suitable for PLA structures, which decreases the switching activity by disabling the inputs of the passive sub-FSMs. Another method is to gate the global clock signal in order to power down the passive sub-FSMs. In [5] a method for identifying self-loops and introducing gating of the clock, in cases where this leads to power savings, was developed. The extension of this work [6-8] has resulted in a synthesis procedure that includes automatic partitioning of the sub-FSMs. The partitioning is here performed before state assignment. Another approach for data flow dominating designs, presented by Hwang et al. [12], is to partition both the controller as well as the data path in to separate partitions. That is, each partition will have a controller part and a data path. Based on their own power estimations, ignoring clock distribution, they claim large power savings although this approach does not use clock gating.

For FSMs without, or with few self-loops, e.g. counters, it is possible to detect smaller FSMs that have self-loops if the partitioning is taking place after state assignment [9] and reduction in the power consumption can also be made for these.

A problem with the approach above is that the large power overhead dissipated in the circuitry controlling the clock-gating. This prevents small partitions and thus low power dissipation. In this paper we propose a low-power asynchronous communication controller for the interaction between the sub-FSMs. These are controlling the gating of the global clock. We call the circuit *Clock Controller Block* (CCB). The asynchronous CCBs are inter-changeable with the synchronous CCBs that are used in [8].

0-7803-5682-9/99/$10.00©1999 IEEE.

3. ASYNCHRONOUS CONTROL OF GATED-CLOCK FSMs

The proposed structure for asynchronous control of clock-gated FSMs is depicted in figure 1. A de-composed FSM consists of (1) a number of sub-FSMs (partitions), (2) an equally large number of asynchronous CCBs, (3) nand-gates for gating the local clocks, and (4) one inverter for the global clock signal.

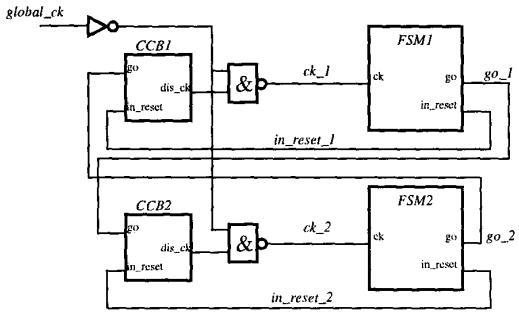

Fig. 1: Asynchronous control of gated-clock FSMs with two partitions

Here the CCB is an asynchronous finite-state machine (AFSM). The advantage of using an AFSM is that it does not require a clock signal for synchronization. In our case it means that passive CCBs are not clocked at all, while the synchronous CCBs are always clocked whether they are active or not. The signal interface is basically the same for the synchronous and asynchronous CCB. However, for the asynchronous CCB there are two environmental requirements that must be fulfilled. This will be discussed later in detail.

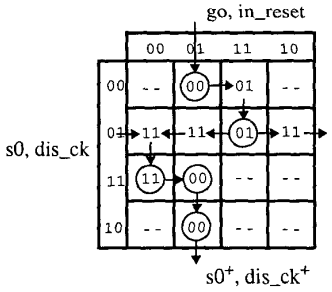

Fig. 2: The asynchronous behaviour in a transition map

The CCB is fairly simple and requires only one state variable and can be described in, and synthesized from, a transition diagram. Its behaviour is fully described in figure 2 with the input signals *go* and *in_reset*, output signal *dis_ck*, and internal state variable *s0*.

The CCB is activated when there is a transition from

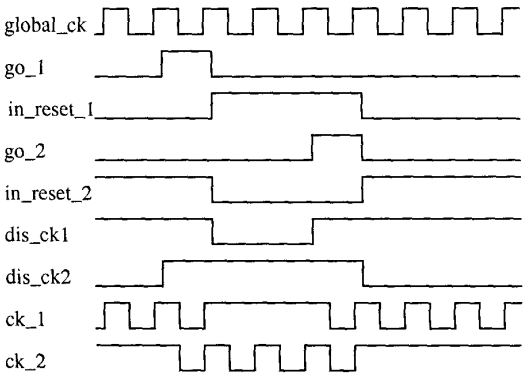

Fig. 3: Timing diagram for transition from FSM1 to FSM2

one sub-FSM to another. This is illustrated by the timing diagram in figure 3. Initially the FSM1 is active and FSM2 is de-activated, which is indicated with the signals *in_reset_1* = '0' and *in_reset_2* = '1'. At the point when FSM1 hands over to FSM2 it submits a positive transition on signal *go_1*. In the next clock cycle FSM1 goes to a *reset state* and becomes de-activated, *in_reset_1* goes '1', and FSM2 becomes activated, *in_reset_2* goes '0'. From the sequence described above we can see that the transitions on the CCB input signals are always separated by at least one clock cycle, except for the case where we have a falling edge of *go* and *in_reset*. Handling of this multiple input change for these signals is described in the transition map starting from the state (*s0,dis_ck,go,in_reset*) = (0,1,1,1). In all other cases it is possible to design the CCB under the fundamental mode assumption as long as the clock cycle is larger than the settling time for the CCB. The second requirement is that the *go* signal from the sub-FSM must be hazard-free.

In general, a sub-FSM can be activated by one of many sub-FSMs. A multi-input CCB is constructed by taking the *or*-function of all outputs of one-bit CCBs, see figure 4.

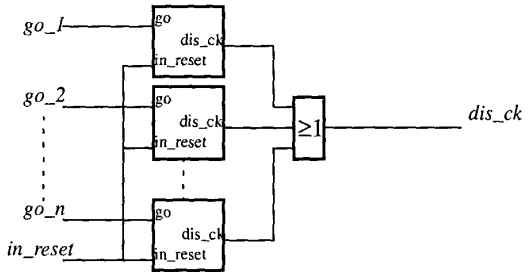

Fig. 4: Multi-input asynchronous CCB

4. EXPERIMENTAL RESULTS

We demonstrate the effectiveness of the asynchronous CCB in comparison to its synchronous counterpart through a case-study. For the comparison a binary counter with 256 states has been selected. Beside from the un-partitioned 256 state counter, seven different versions of the counter have been constructed from equal sized sub-FSMs. The original 256 state counter is partitioned into 2, 4, 8, 16, 32, 64 and 128 partitions (sub-FSMs) of equal sizes. These sub-FSMs are used in two sets of experiments. In the first experiment the synchronous CCB, proposed in [6], for interaction between sub-FSMs is employed. In the second experiment the asynchronous CCB is used. All FSMs, asynchronous CCBs, and synchronous CCBs have been synthesized from VHDL RTL-descriptions to gate-level using *Design Compiler* (Synopsys Inc). The technology is 0.5 μm CMOS and the operating conditions are set to *worst-case military* with a clock frequency of 125 MHz. Under these conditions the 256 state counter just reaches the targeted clock frequency. By pushing the circuit to the limit it is possible to see the positive effects of using many simple sub-FSMs compared to one large FSM. Contrary to small FSMs, the large FSM will increase both in size and power dissipation in *next-state* and *output* logic in order to reach hard speed requirements.

The power estimates were obtained from *DesignPower* (Synopsys Inc.) that uses zero-delay simulation to propagate the switching activity throughout the design. Setting values for transition probabilities for the input signals is not an issue since the counter only has reset and clock inputs. In the power estimates the following factors have been taken into account; (1) internal cell dissipation, (2) dissipation in signal interconnection capacitance, and (3) dissipation in clock distribution nets (both for local gated clock and global clock).

The additional control circuitry that handles the interaction between the sub-FSMs will introduce additional power dissipation (power overhead). The number of CCBs are equal to the number of partitions. Only one of the CCBs is active at a time except for one cycle where a transition from one sub-FSM to another occurs. Here two CCBs are active. A CCB has three operational modes:

- **Hand-over**: when a transition from one sub-FSM to another takes place. In this mode is the asynchronous CCB controller circuit is active and responds to the go-signal. The power consumption is 35% higher for the synchronous CCB.

- **Enable**: the CCB is passive and enables the local clock signal to the sub-FSM. In this mode the asynchronous CCB controller circuit is passive and do not dissipate any power. The contribution to the power consumption comes from the switching av the clock input and output of the nand-gate that is used for clock-gating. The power consumption is 409% higher for the synchronous CCB.

- **Disable**: the CCB is passive and disables the local clock signal (the clock is gated). The power consumption comes from switching the input of the nand-gate. The power consumption is 583% higher for the synchronous CCB.

A CCB dissipates power both when it is active ($P_{CCB,hand\text{-}over}$) and when it is passive ($P_{CCB,enable}$, $P_{CCB,disable}$). When the number of partitions increase, the increase in power overhead will mainly be caused by $P_{CCB,disable}$ and the higher probability of transitions between different sub-FSMs (two CCBs are then simultaneously active). In addition, the total clock load will grow and it will lead to more clock buffers. For optimal buffering [10], it means that switching capacitance in the clock buffers will be 40% of the capacitance they are driving [11]. With asynchronous CCBs that have lower power dissipation, see table 1, the power overhead compared to the synchronous counterpart can be reduced, in particular $P_{CCB,disable}$. The asynchronous solution is better suited to be used in a clock-gating scheme since it consumes little power when it is passive.

Table 1. Power consumption in CCBs [μW]

CCB type	$P_{CCB,hand\text{-}over}$	$P_{CCB,enable}$	$P_{CCB,disable}$
Synch.	275	237	140
Asynch.	203	58	24

From figure 5 it is clear that the asynchronous CCB will introduce less power overhead, especially when the number of partitions is large.

When reducing the power overhead, the cost for partitioning a FSM decreases. To set the power overhead in relation to the power dissipated inside the sub-FSMs, the total power consumption for the 256 state binary counters with different number of partitions is plotted in figure 6. For counters with asynchronous CCBs a minimum in the total power consumption has been achieved with 8 partitions. With synchronous CCBs the un-partitioned counter gives the lowest power consumption.

5. CONCLUSIONS

In this paper we have shown that asynchronous control of synchronous sub-FSMs is of great importance for the reduction of power consumption in de-composed FSMs. The asynchronous CCB has lower switching capacitance attached to the global clock signal. When a FSM is divided into many partitions, the capacitive load on the clock net, introduced by the clock control blocks, will make a significant contribution to the total power con-

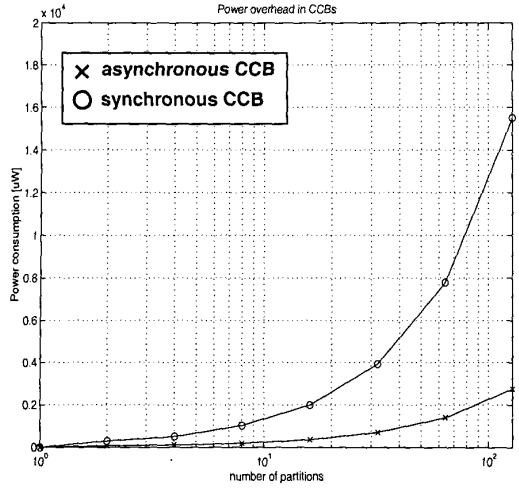

Fig. 5: Power overhead in CCB's

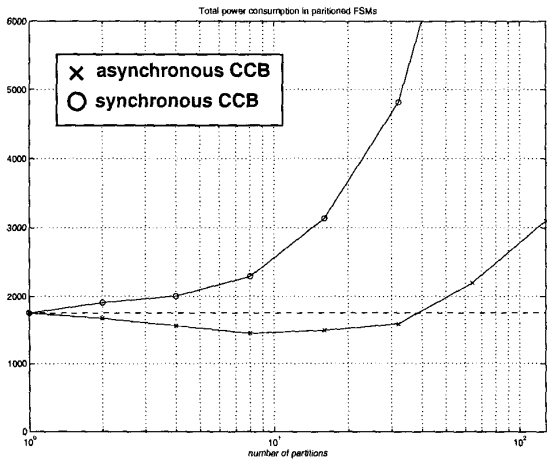

Fig. 6: Total power consumption in FSMs.

sumption. By lowering this overhead we are able to make fine grained de-composition of the FSM. For the binary counter we used for the case-study in this paper, a reduction in power consumption was obtained for up to 32 partitions using the asynchronous CCB. When using the synchronous controller no power reduction could be observed. Finally, it should be noted that a binary counter is a *worst-case* type of design where the potential of using the de-composition technique is small. This technique, using either synchronous or asynchronous control, will result in larger power reductions for FSMs with idle loops [5] or where there is a high probability of state transitions among a small number of states during a limited period of time [8].

6. REFERENCES

[1] D. Liu and C. Svensson, Power Estimation in CMOS VLSI Chips, *IEEE Journal of Solid State Circuits*, vol. 29, no. 6, June 1994

[2] Anatha P. Chandrakasan, Samuel Sheng, and Robert W. Brodersen, Low Power CMOS Digital Design, *IEEE Journal of Solid State Circuits*, vol. 27, no. 4, Apr. 1992, pp. 473-484

[3] Jose Monteiro, Srinivas Devadas, and Abhijit Ghosh, Sequential Logic Optimization for Low Power Using Input-Disabling Precomputation Architectures, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 3, March 1998

[4] Aurobindo Dasgupta and Shantanu Ganguly, Divide & Conquer: A Strategy for Synthesis of Low Power Finite State Machines, *Proceedings of the International Conference on Computer Design. VLSI in Computers and Processors*, 1997, pp. 740-745

[5] Luca Benini, Polly Siegel, and Giovanni De Micheli, Saving Power by Synthesizing Gate Clocks for Sequential Circuits, *IEEE Design and Test of Computers*, vol. 11, Dec. 1994, pp. 32-41

[6] Luca Benini and Giovanni De Micheli, Transformation and Synthesis of FSMs for low-power gated-clock implementation, *Proceedings of International Symposium on Low Power Design. ACM*, New York, NY, USA, 1995 pp. 21-26

[7] Luca Benini and Giovanni De Micheli, Automatic Synthesis of Low-Power Gated-Clock Finite-State Machines, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 6, 1996, pp. 630-643

[8] Luca Benini, Frederik Vermeulen, and Giovanni De Micheli, Finite-State Machine Partitioning for Low-Power, *Proceedings of the IEEE International Symposium on Circuits and Systems* 1998, vol. 2, pp. 5-8

[9] Sumit Roy, Prithviraj Banerjee, and Majid Sarrafzadeh, Partitioning Sequential Circuits for Low Power, *Proceedings of the 11th International Conference on VLSI Design*, IEEE Comput. Soc., Los Alamitos, CA, USA, 1997, pp. 212-217

[10] Carver Mead and Lynn Conway, *Introduction to VLSI systems*, Addison-Wesley, 1980

[11] N. Hedenstierna and K.O. Jeppson, CMOS circuit speed and buffer optimization, IEEE Transaction on Computers, vol. 35, 1986, pp. 880-895

[12] Enoch Hwang, Frank Vahid, Yu-Chin Hsu, "FSMD Functional Partitioning for Low Power", *Proceedings of Design Automation and Test in Europe (DATE)*, March 1999, pp. 22-28.