Individual Flip-Flops with Gated Clocks for Low Power Datapaths

Tomas Lang, Enric Musoll, and Jordi Cortadella

***Abstract*—Energy consumption has become one of the important factors in digital systems, because of the requirement to dissipate this energy in high-density circuits and to extend the battery life in portable systems such as devices with wireless communication capabilities. Flip-flops are one of the most energy-consuming components of digital circuits. This paper presents techniques to reduce energy consumption by individually deactivating the clock when flip-flops do not have to change their value. Flip-flop structures are proposed and selection criteria given to obtain minimum energy consumption. The structures have been evaluated using energy models and validated by switch-level simulations. For the applications considered, significant energy reductions are achieved.**

***Index Terms*—Flip-flop energy model, gated clocks, low power datapaths.**

I. Introduction

THE REDUCTION of energy consumption has become one of the important factors in digital systems, because of the requirement to dissipate this energy in the high-density circuits that are possible with submicron technology, and of extending the battery life in portable systems, such as high-performance portable computers and personal digital assistants (PDA's) with multimedia and wireless communication capabilities.

The design of low power circuits can be tackled at different levels, from system to technology, as illustrated for instance in [4], where numerous references to the topic can be found. We here concentrate on techniques at the logic level for CMOS technology, aiming at reducing the average energy consumed in the datapath registers during an operation. This energy accounts for a large fraction of the total energy of the system. For instance, for the PVQ decoder described in [12] the energy consumed by the registers is 90% of that consumed by the datapath, and this value is of about 75% for the radix-4 multiplier and for the accumulator that we have implemented. Double-edge triggered (DET) flip-flops [8] have been proposed as one of the techniques that can produce significant energy savings (20%) for this type of systems. However, the synchronization model imposed by DET flip-flops is not always applicable. For this reason, we focus on the more conventional single-edge triggered (SET) flip-flops.

In this paper we consider the approach of disabling the clock when the flip-flop must not change, which reduces the energy consumed by the clock circuits internal to the flip-flop. This approach can also be combined with appropriate data representations to reduce the switching activity [10]. This disabling technique is related to methods proposed to shutdown inactive portions of the system [4], [6], such as inhibiting the register load and/or inhibiting the clock (gated clock). Inhibiting register load is proposed in [1] but this does not reduce the effective load on the clock. The use of gated clocks to reduce the activity of logic modules is described in [3]–[5], [12], [13].

Our proposal is at a finer level of granularity in which individual flip-flops are activated/deactivated according to their local behavior. Flip-flop structures are proposed and models for energy consumption are developed, resulting in criteria to select the most-appropriate structure depending on the flip-flop activity.

To validate the models, the circuits have been implemented in a Sea-of-Gates design framework [7] and the energy consumption has been determined using a switch-level simulator [14] based on an RC model with timing.

We conclude that a significant reduction in energy can be obtained with the techniques proposed in this paper.

II. An Energy Model for D Flip-Flops

We now discuss a model for the energy consumed by a D flip-flop (called in the sequel ffconv), which is used as a framework and a reference to compare with the proposed structures. As shown in Fig. 1(a) the input D is the output z of a combinational network and the input CLK comes directly from the clock. Because of the structures discussed later, we use a trailing-edge triggered flip-flop.

Since in CMOS the main component of the energy is dynamic, the flip-flop consumes energy whenever there is a transition in any of its inputs, that is

1) during the transitions of z. These transitions are of two types: the real transition corresponding to the function implemented by the combinational network, and spurious transitions (glitches);
2) during both edges of the clock.

Manuscript received March 24, 1997. This work was supported in part by the National Science Foundation Grant MIP 9314172 and CICYT Grant TIC 95-0419. This paper was recommended by Guest Editors S. Kiaei and E. G. Friedman.

T. Lang is with the Department of Electrical and Computer Engineering, University of California, Irvine, CA 92697 USA.

E. Musoll was with the Department of Computer Architecture, Universitat Politècnica de Catalunya, 08071 Barcelona, Spain. He is now with National Semiconductor Corporation, Santa Clara, CA 95054 USA.

J. Cortadella is with the Department of Computer Architecture, Universitat Politècnica de Catalunya, 08071 Barcelona, Spain.

Publisher Item Identifier S 1057-7130(97)04785-X.

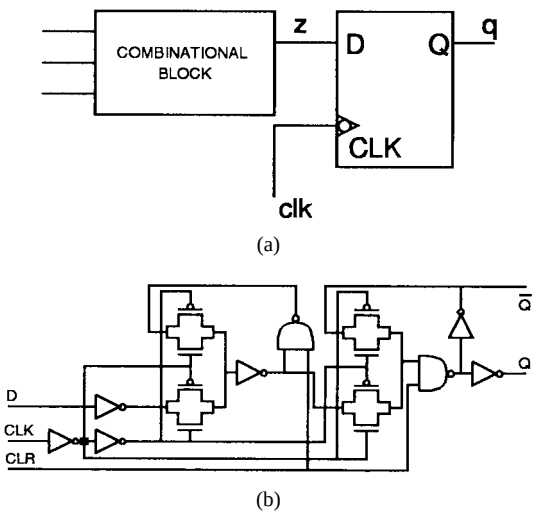


Fig. 1. (a) Trailing-edge D flip-flop (ffconv). (b) A possible implementation.

Moreover, the energy consumed during a transition depends on the present and next states of the flip-flop and on the relative timing of the transitions, so that the following situations are identified.

- With respect to the state, we consider three cases, namely, when the flip-flop changes, when the state remains 0, and when the state remains 1.
- With respect to the relative timing of the z transitions and the clock leading edge, we consider two cases, namely, when the transition occurs before and after this edge.

Consequently, we obtain the following parameters for the energy consumed by one event of the corresponding type:

E_c Flip-flop changes. This includes the energy consumed during clock edges and during the real transition of z.
E_0 Flip-flop remains in state 0 (no transition in z).
E_1 Flip-flop remains in state 1 (no transition in z).
E_{gb} Glitch of z before the leading edge of the clock.
E_{ga} Glitch of z after the leading edge of the clock.

Note that a glitch corresponds to two spurious transitions.

The average energy per cycle is obtained by the summation of the products of the energy parameter times the corresponding average number of events per cycle (denoted by N_x). That is

$$E = E_c N_c + E_0 N_0 + E_1 N_1 + E_{gb} N_{gb} + E_{ga} N_{ga}$$

with $N_c + N_0 + N_1 = 1$ and $N_{gb}, N_{ga} \geq 0$.

A D flip-flop such as the one depicted in Fig. 1(b) has been implemented based on the design in [11] and simulated at the switch level. We have obtained

$$E = (14.6 + 0.3L)N_c + 7.6(N_0 + N_1) + 1.8N_{gb} + 7.8N_{ga}$$

where L is the fanout[1] and the unit corresponds to the energy consumed by one output transition of a 2-input NAND gate with unit load. This expression shows that a significant fraction of the energy is consumed when the flip-flop output does not change $(7.6(N_0 + N_1) + 1.8N_{gb} + 7.8N_{ga})$.

[1] A unit load corresponds to the input capacitance of an inverter.

III. Clock Activation

To reduce the energy consumed when the flip-flop output does not change we control the CLK input of the flip-flop by the *activation signal* A so that

$$CLK = A \cdot clk$$

This is implemented by the AND gate shown in Fig. 2(a). This gate can be easily integrated in the flip-flop realization by adding two transistors. In the sequel, we call this a gated flip-flop (henceforth, ffg). Moreover, in the ffg the D input is not necessarily z (as in ffconv), since when $A = 0$ the value of D is a don't care. We call this input function d.

As shown in Fig. 2(b), for the ffg to operate correctly it must be trailing-edge triggered and the signal A must not have negative transitions while the clock is high. For instance, if in cycle T_1 signal A has a negative transition while clk is high, then a trailing edge in clk_g is produced, changing q to 0 incorrectly. Consequently, it must be assured that the negative transition of A (including any possible glitches) occurs before the clock pulse. The effect of this on the delay of the network is considered in Section V. If a leading-edge flip-flop is used instead, an OR gate should replace the AND and similar restrictions exist on the activation signal. In the sequel, we only consider the trailing-edge case.

The activation signal A is a function of z and Q. For the network to operate correctly, it is necessary that $A = 1$ whenever the flip-flop has to change value. This condition is stated by the following Boolean inequality:

$$A \geq z \oplus Q. \tag{1}$$

Moreover, for a correct operation, the input d must have a value such that

$$d = \begin{cases} z & \text{if } A = 1 \\ \text{don't care} & \text{if } A = 0 \end{cases}$$

These conditions are expressed by the following Boolean inequality

$$z \cdot A \leq d \leq z + \overline{A}. \tag{2}$$

The actual functions A and d should satisfy conditions (1) and (2), reduce the energy consumption, and satisfy other requirements, such as delay and area.

In the next two sections we consider structures that satisfy (1) and (2) for two classes of situations, as follows.

Independent Case The combinational network does not include Q as an input. That is

$$z = f(x_1, x_2, \cdots, x_k) \tag{3}$$

where $\underline{x} = (x_1, x_2, \cdots, x_k)$ is the input of the network.

Dependent Case Q is an input to the combinational network, namely

$$z = f(x_1, x_2, \cdots, x_k, Q) \tag{4}$$

As shown in Section III-B this dependence can be used to simplify the network to obtain A.

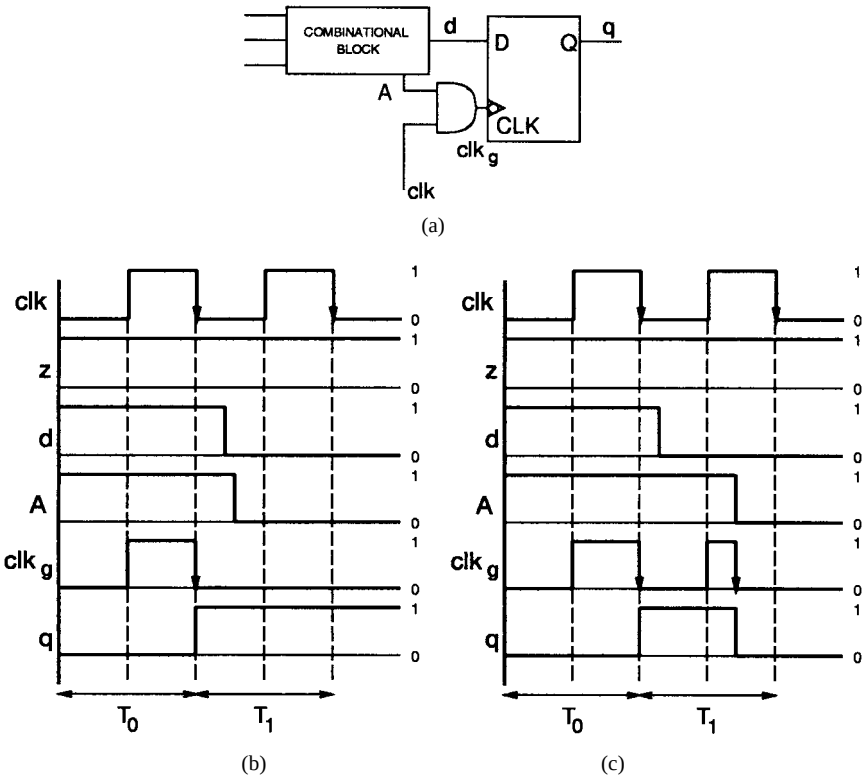


Fig. 2. (a) Gated trailing-edge flip-flop (ffg); example of correct (b) and incorrect (c) timing.

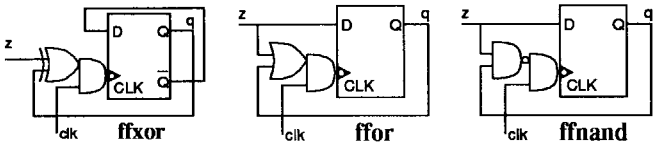
A. Independent Case

For the independent case of expression (3), we now consider the three structures shown in Fig. 3. For the ffxor structure, $A = 1$ only when the flip-flop output has to change, so that the flip-flop clock is never activated when it does not change, in contrast with ffconv in which the clock is activated in every cycle. Although this corresponds to the minimum clock activation required, this structure might not be the most appropriate when the extra energy, area, and delay of the XOR gate is included. Because of this, we consider also the other two structures, in which the complexity of the generation of the activation signal is reduced at the expense of activating the clock in some cycles when the flip-flop output does not change. As shown in the Table of Fig. 3, in the ffor case the clock is activated when the flip-flop changes and when it remains in state 1. The implementation is shown in Fig. 3(b); the OR gate is integrated with the AND gate and with an internal inverter in a complex gate. Similarly, in the ffnand the clock is activated when the flip-flop changes as well as when it remains in state 0.

The table also shows for each structure the corresponding expression for A, the condition of (2), and the simplest function that satisfies this condition.

1) Energy models: We now consider energy models for the new flip-flop structures and give the values of the energy parameters obtained from the switch-level simulations. The expression for the energy is similar to that of Section II, except that now, as discussed in Section III, all transitions of z have to occur before the leading edge of the clock. The resulting expression is

$$E = E_c N_c + E_0 N_0 + E_1 N_1 + E_{gb} N_{gb}.$$



	$0 \Leftrightarrow 1$	$0 \Rightarrow 0$	$1 \Rightarrow 1$	A	Eq. (2)	d
ffconv	y	y	y	1	$z \le d \le z$	z
ffxor	y	n	n	$z \oplus Q$	$z\overline{Q} \le d \le z + \overline{Q}$	$\overline{Q}$
ffor	y	n	y	$z + Q$	$z \le d \le z + \overline{Q}$	z
ffnand	y	y	n	$\overline{zQ}$	$z\overline{Q} \le d \le z$	z

(d)

Fig. 3. Flip-flop structures, output transitions for which the clock is activated, and Boolean functions for the inputs.

The parameters for flip-flop structures that have been implemented are shown in Table I (the table also gives relative area and delay, from D to Q, one unit load). From this table we obtain the following expected conclusions.

- For ffxor the parameters E_0 and E_1 are small, since the clock is deactivated in these cases. On the other hand, the parameters corresponding to transitions in z (that is E_c and E_{gb}) increase with respect to ffconv, because of the XOR gate.
- For ffor, the parameter E_0 is small since the clock is deactivated in this case. E_c and E_{gb} increase somewhat with respect to ffconv, but less than for ffxor. For ffnand, the situation is similar to ffor, exchanging 0 and 1.

For specific values of the characteristics of flip-flops (defined by E_x) and the average number of transitions (defined by N_x), the flip-flop structure that consumes the least energy

TABLE I
PARAMETERS FOR THE IMPLEMENTED FLIP-FLOP STRUCTURES

Flip-flop	E_c	E_0	E_1	E_{gb}	E_{ga}	Relative delay	Relative area
ffconv	14.6+0.3L	7.6	7.6	1.8	7.8	1	1
ffxor	19.8+0.3L	0.8	1.0	6.8	-	1.2	1.5
ffor	15.6+0.3L	0.8	8.4	2.6	-	1.1	1.1
ffnand	16.6+0.3L	8.0	1.0	4.2	-	1.1	1.2

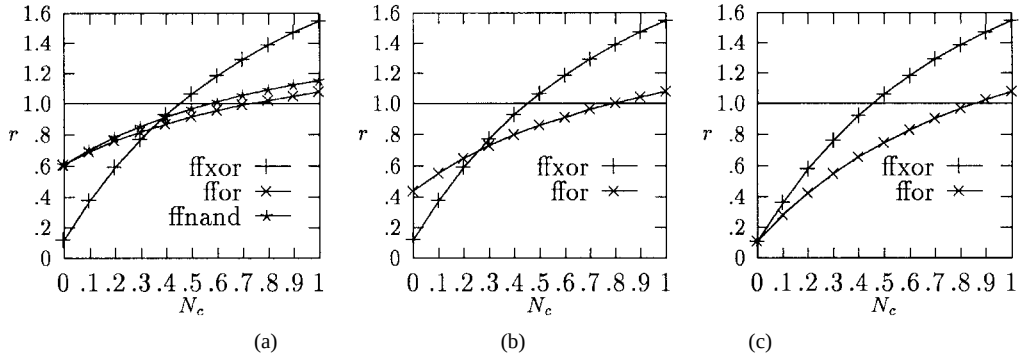


Fig. 4. Energy ratios for (a) $N_0 = N_1$. (b) $N_0 = 2N_1$. (c) $N_1 \approx 0$.

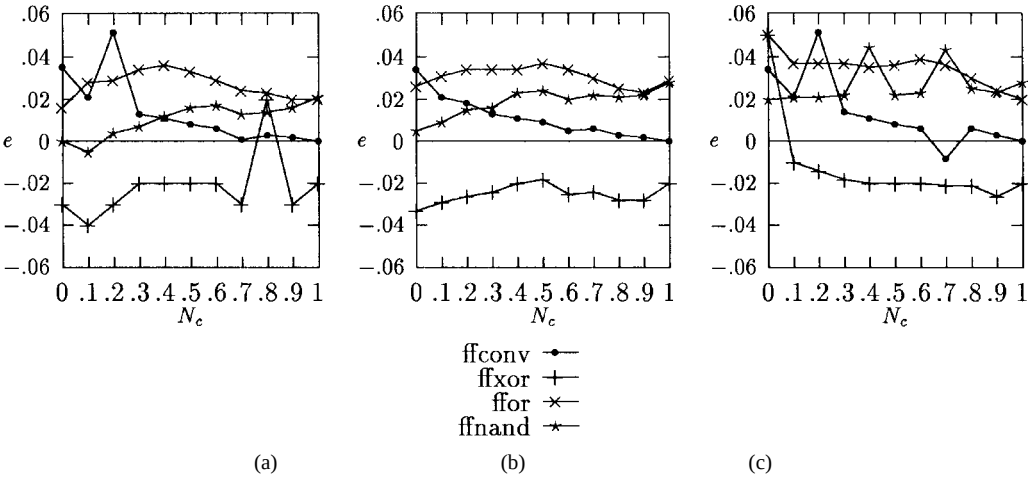


Fig. 5. Relative error between model and simulations. (a) $N_0 = N_1$. (b) $N_0 = 2N_1$. (c) $N_1 \approx 0$.

is determined. Note that it might be convenient to use different types of flip-flops in the same register, since the average number of transitions might be different.

To compare the energy of different structures, we use the ratio of energy with respect to a conventional implementation, that is $r = E/E_{\text{conv}}$ so that smaller r corresponds to larger reductions. Fig. 4 shows this ratio for several specific cases, a fanout of unit load, and no glitches. The ratio is given as a function of N_c for different values of $N_1/N_0 \leq 1$. Since for these values, the ffor structure is better than the ffnand structure we do not show the ffnand case; for $N_1/N_0 > 1$ similar plots would be obtained with the ffnand structure better than the ffor structure. We conclude that, the reduction is larger for smaller values of N_c; the best flip-flop depends on N_c: for small N_c the best is ffxor, then for intermediate N_c the best is either ffor or ffnand, and finally, for large N_c the best is ffconv. Moreover, the breakpoint between regions depends on the ratio N_0/N_1. The case $N_1 \approx 0$ corresponds to a limiting situation, showing the best reduction that can be achieved.

We have not considered N_g in these plots. The value of N_g depends on the particular application and on the design and implementation approach. In those applications in which energy consumption is a crucial design factor, combinational blocks might be designed to reduce the number of glitches [2], [9]. In the applications considered in this paper, we have measured the glitches for the specific implementation and have included their effect in the energy calculations.

2) Simulation of Implementations: We have implemented the flip-flop structures and obtained the energy from switch-level simulations. As can be seen from Fig. 5, the relative error of the simulations with respect to the models ($e = (\text{meas.} - \text{model})/\text{meas.}$) is less than 5%.

B. Dependent Case

We now consider the dependent case of (4). Because of the dependence of z on Q, instead of first obtaining z and then A and d satisfying (1) and (2), the signals A and d are obtained directly. Consequently, we use the ffg structure of Fig. 2(a). To obtain the energy expression we need to identify the different possible events. They are

- the flip-flop output changes. In this case $A = 1$. We use the subscript c for this event;
- the flip-flop output does not change and $A = 0$ (clock deactivated). We use the subscript $A0$ for this event;
- the flip-flop output does not change but $A = 1$. We use the subscript $A1$ for this event.

Moreover, there are two types of possible glitches: in input A and in input d. The energy expression is then

$$E = E_c N_c + E_{A0} N_{A0} + E_{A1} N_{A1} + E_{gA} N_{gA} + E_{gD} N_{gD}.$$

Our implementation of this flip-flop results in

$$E = (14.5 + 0.3L)N_c + 0.8N_{A0} + 8.0N_{A1} + 0.8N_{gA} + 1.8N_{gD}.$$

As indicated in Section II, the conditions for A and d are

$$A \geq z \oplus Q$$
$$z \cdot A \leq d \leq z + \overline{A}.$$

IV. Applications

We now illustrate the use of the flip-flop structures in both the independent and dependent cases. Two detailed examples and a summary of the results reported for other examples are presented. Further information on the implementation details of all the examples can be found in [10].

In the evaluations we use flip-flops with the characteristics described before, considering an external load for the flip-flops of two unit loads. In the sequel, L represents the total load (external and internal) of the flip-flops. The internal load depends on the implementation chosen for the application. The comparisons are in terms of the ratio of the energy of the conventional and the low power implementations. For each application, the energy has been estimated in two different ways, namely, (1) by applying the energy model previously presented, and (2) by means of switch-level simulations [14] of a realization of the circuit. To compute the average, we have performed a suitable number of simulations with inputs obtained from specific distributions, as indicated in each application. Moreover, we have also implemented the required combinational network in a Sea-of-Gates design style [7] and report the energy ratio of the overall circuit obtained from the switch-level simulations.

The applications in this paper have a bit-width of 16 bits. Similar results are expected for larger operand bit-widths.

A. Independent Case: Sampling of Signals

This example corresponds to systems in which a signal is sampled and the value is stored in a register. To reduce the energy consumed by the register, special flip-flop structures are advantageous for the bits that have low activity. This situation happens, for example, for the following two scenarios.

- Slow-varying signals (compared with the sampling rate). This occurs, for instance, in the sampling of audio and video signals.
- Signals in which the values are most of the time close to a fixed point. This occurs, for instance, in systems that monitor physical quantities, such as temperature and pressure.

As an illustration, we consider the system shown in Fig. 6(a), in which a triangular signal is sampled and the value loaded into a 16-bit register [Fig. 6(b)]. The frequency of the signal and the sampling period are such that the probability of change of flip-flop i is 2^{-i} $(0 \leq i \leq 15)$ as shown in Fig. 6(c). This is due to the fact that the value in the register changes by ± 1 each cycle. Similar results should be obtained for other slow-varying signals.

Conventional Implementation

A conventional implementation would use conventional D flip-flops for all bits. Assuming that the input signal has no glitches, the energy is

$$E = 2E_c + 14 \times 0.5(E_0 + E_1) = 136.8$$

since, on the average, two flip-flops change $(\Sigma_{i=0}^{15} 2^{-i} \approx 2)$.

Low Power Implementation

For the flip-flop transitions we have $N_0 = N_1 = 0.5(1 - N_c)$ and $N_c = 2^{-i}$ for flip-flop i. According to the conclusions obtained before we use the following flip-flop structures.

- Flip-flop 0: $N_c = 1$, use ffconv, $E = 15.2$
- Flip-flop 1: $N_c = 0.5$, use ffor, $E = 10.4$
- Flip-flops 2 to 15 $N_c \leq 0.25$, use ffxor,

$$E = 0.5E_c + 13.5(0.5)(E_0 + E_1) = 22.4\left(\sum_{i=2}^{15} 2^{-i} \approx 0.5\right).$$

The total energy is 48.0 units and the ratio is $r = 48.0/136.8 = 0.35$. We have implemented both register designs and, from the switch-level simulation, have obtained a ratio of 0.32.

Since the ffxor structure has larger area and delay than the ffor structure we evaluate the energy when ffor structures are also used for flip-flops 2 to 15. The energy is now

- Flip-flop 0: $N_c = 1$, use ffconv, $E = 15.2$
- Flip-flop 1 to 15: use ffor,

$$E = 0.5E_c + 14.5(0.5)(E_0 + E_1) = 74.8.$$

Now the total energy is 90.0 units and the ratio is $r = 90.0/136.8 = 0.66$. For this case, the switch-level simulation produces a ratio of 0.71.

B. Dependent Case: Accumulator

The operation is described by the recurrence

$$z[i+1] = z[i] + y[i]$$

where $z[0] = 0$ and $y[i]$ are the input operands.

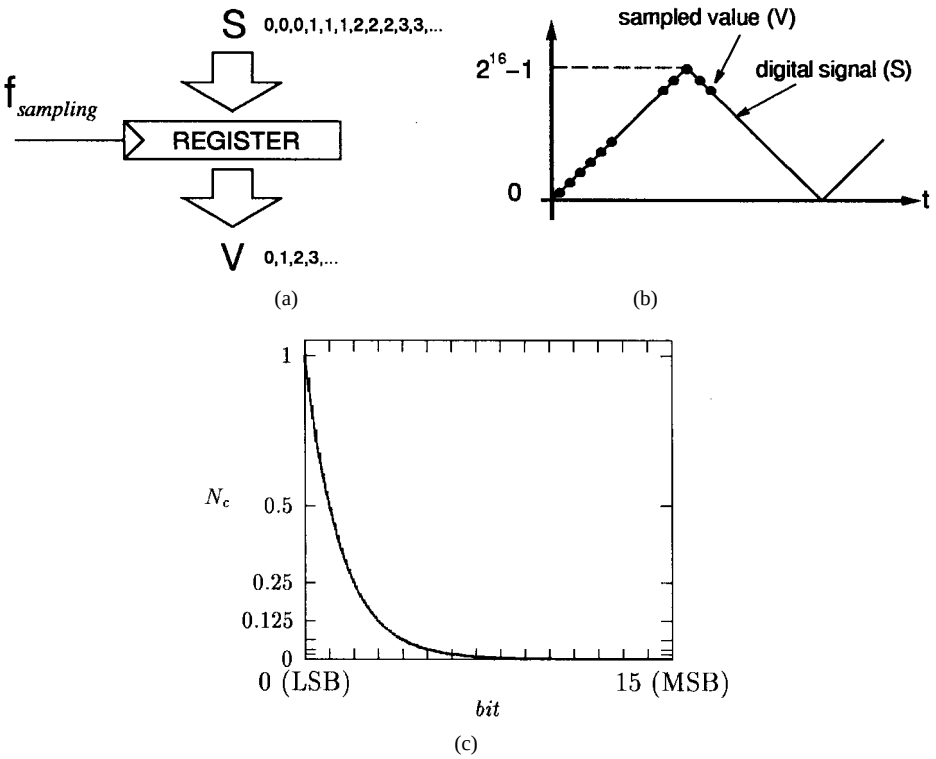


Fig. 6. Sampling of a slow-varying (triangular) signal. (a) System. (b) Signal. (c) Activity of register bits.

We consider two types of representations for $z[i]$: nonredundant radix-2 and carry-save. Moreover, we consider the case in which the value of $y[i]$ is uniformly distributed; similar analyzes would be applicable to other distributions.

1) Nonredundant Representation

For this case a carry-propagate adder is used. That is, as shown in Fig. 7(a), the recurrence is implemented by the following Boolean expression:

$$z_j[i+1] = z_j[i] \oplus y_j[i] \oplus c_j[i]$$

where $c_j[i]$ is the jth bit of the carry-vector. This vector can be produced in several ways, such as using carry-ripple or carry-lookahead structures.

Conventional Implementation

In this case conventional flip-flops are used. If the values of $y[i]$ are uniformly distributed, the average number of transition in z_j is $N_c = 0.5$. We use a symmetric clock in which case the glitches occur before the leading edge of the clock since the average length of the carry chains is small with respect to the worst-case delay of the adder. We have measured $N_{gb} = 0.52$. Consequently, the energy per flip-flop is (for $L = 4$)

$$E = 0.5E_c + 0.5(0.5)(E_0 + E_1) + 0.52E_{gb} = 12.6.$$

Implementation with Activation Signal

In this case a ffg is used and the simplest activation signal is

$$A_j[i] = z_j[i+1] \oplus z_j[i] = y_j[i] \oplus c_j[i]$$

Consequently, instead of having to include an additional XOR gate in the implementation of A, this results in a reduction of one XOR gate with respect to z_j, as shown in Fig. 7(b). In addition

$$d = \overline{Q}.$$

Since in this case A is always 0 when the flip-flop output does not change, we have $N_{A1} = 0$. We have measured $N_c = 0.5$, $N_{A0} = 0.5$, and $N_{gA} = 0.65$, resulting in the energy per ffg (for $L = 4$)

$$E = 0.5E_c + 0.5E_{A0} + 0.65E_{gA} = 8.8$$

The ratio is $r = 8.8/12.6 = 0.70$.

2) Carry-Save Representation

To have a faster operation redundant adders are used. We illustrate the use of a carry-save adder in which the accumulated value is represented by the sum of two vectors so that $z = s + c$. The expressions are

$$s_j[i+1] = s_j[i] \oplus c_j[i] \oplus y_j[i]$$
$$c_{j+1}[i+1] = Maj(s_j[i], c_j[i], y_j[i]).$$

Conventional Implementation

A bit slice consists of one full adder and two conventional flip-flops, as shown in Fig. 8(a). In this case, using a symmetric clock makes the glitches occur after the leading edge of the clock since the average delay is similar to the worst-case delay. We have measured the following frequencies.

- For $s, N_c = 0.50, N_{ga} = 0.37$; For $c, N_c = 0.66, N_{ga} = 0.20$;

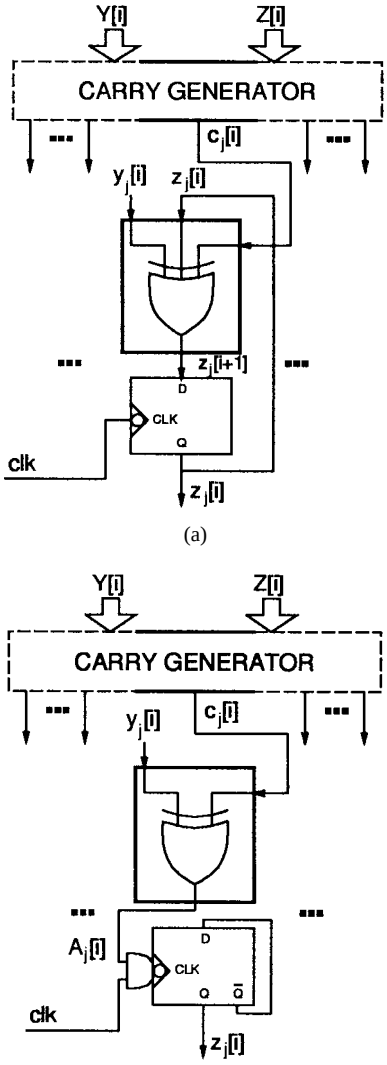


Fig. 7. Nonredundant accumulator (a) with ffconv and (b) with ffg.

resulting in the energy per bit-slice (for $L = 4$):

$$E = 1.16E_c + 0.84(0.5)(E_0 + E_1) + 0.57E_{ga} = 29.2$$

Implementation with Activation Signal

The activation signal for s is the same as that for z in the previous case, that is

$$A_{s_j} = c_j[i] \oplus y_j[i].$$

Because of the "shifting" in the carry vector, the carry is of the "independent" type (as in Section III-A). We have measured

- For $s, N_c = 0.5$ and $N_{gA} = 0.54$. Moreover, since when the flip-flop does not change A is always 0, we have $N_{A1} = 0$ and $N_{A0} = 0.5$.
- For $c, N_c = 0.66, N_0 = 0.19, N_1 = 0.15, N_{ga} \approx 0$.

Because of these values the best structure for the carry bit is ffconv, as shown in Fig. 8(b). The energy per bit-slice (two

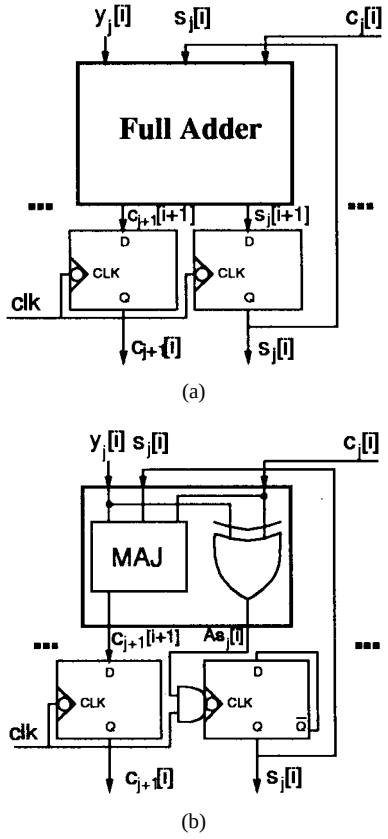


Fig. 8. Carry-save accumulator (a) conventional. (b) With activation signal.

flip-flops) is (for $L = 4$)

$$E = (E, \text{bit } s) + (E, \text{bit } c) = (0.5E_c + 0.5E_{A0} + 0.54E_{gA}) + (0.66E_c + 0.34(0.5)(E_0 + E_1)) = 21.7.$$

The ratio is 21.7/29.2 = 0.74.

Implementation and Simulation

We have implemented all the accumulator designs and obtained by switch-level simulation the following ratios for the energy consumed by the flip-flops and for the overall energy for the accumulator.

- Nonredundant representation with activation signal: $r_{\text{ffs}} = 0.66, r_{\text{overall}} = 0.59$.
- Carry-save representation with activation signal: $r_{\text{ffs}} = 0.72, r_{\text{overall}} = 0.64$.

Similar analyzes would be applicable to similar systems, such as counters.

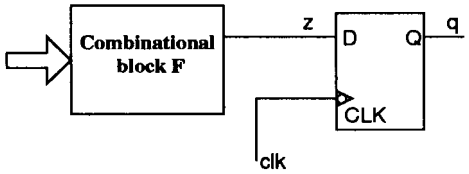
C. Summary of the Applications

Table II summarizes the results obtained from several examples, including those previouly described in this section. For each application we show the case with the smallest overall energy. In the cases of the radix-4 sequential multiplier and the redundant accumulator, a novel encoding technique for redundant data has been also used. Further details on this technique can be found in [10].

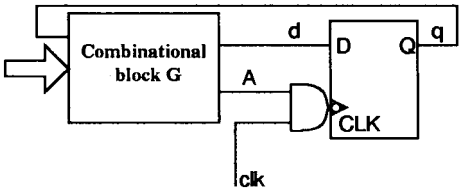
We observe that the simulated results are close to those obtained with the model. We further observe that in those

TABLE II
SUMMARY OF THE APPLICATIONS ($r_{\rm ffs}$ AND $r_{\rm overall}$ ARE THE ENERGY SAVINGS RATIOS OF THE FLIP-FLOPS AND OF THE WHOLE CIRCUIT; I/D STANDS FOR INDEPENDENT/DEPENDENT CASE)

Application	I/D	Model	Measurements	
		r_{ffs}	r_{ffs}	$r_{overall}$
Sampling of signals	I	0.35	0.32	0.32
Radix-4 sequential multiplier	I	0.73	0.76	0.81
(with special encoding)	I	0.63	0.66	0.76
Non-redundant accumulator	D	0.69	0.66	0.59
Redundant accumulator	D	0.74	0.72	0.64
(with special encoding)	D	0.47	0.43	0.74
Masked set-reset	D	0.65	0.68	0.57



(a)



(b)

Fig. 9. Network with (a) conventional flip-flop and (b) gated flip-flop.

applications with combinational logic there is also a significant overall energy reduction.

V. TIMING CHARACTERISTICS

We now consider the increase in cycle time that might result from the use of clock activation. We compare the cycle times of the networks of Fig. 9. We consider the case in which the inputs to the combinational block come from the output of flip-flops triggered with the same clock.

1) Conventional Flip-Flop

The cycle time when a conventional flip-flop is used is determined by the following two restrictions:

1) The delay of z (T_z) plus the set-up time (T_s) and the transmission delay (T_{ff}). That is

$$T_{\rm cyc} \geq T_{ff} + T_z + T_s.$$

2) The restrictions on the minimum width of the two subcycles. That is

$$T_{\rm cyc} \geq T_{\min 0} + T_{\min 1}$$

where $T_{\min 0}$ and $T_{\min 1}$ are the minimum width of the low subcycle and the high subcycle, respectively. These minima depend on the minimum for correct flip-flop operation (T_{pw0} and T_{pw1}) and on the minimum width of the clock, as determined by clock generation and distribution (which we call T_{clk0} and T_{clk1}). So, for example

$$T_{\min 0} = \max(T_{pw0}, T_{clk0}).$$

Consequently,

$$T_{\rm cyc} \geq \max[(T_{ff} + T_z + T_s), (T_{\min 0} + T_{\min 1})]$$

as illustrated in Fig. 10.

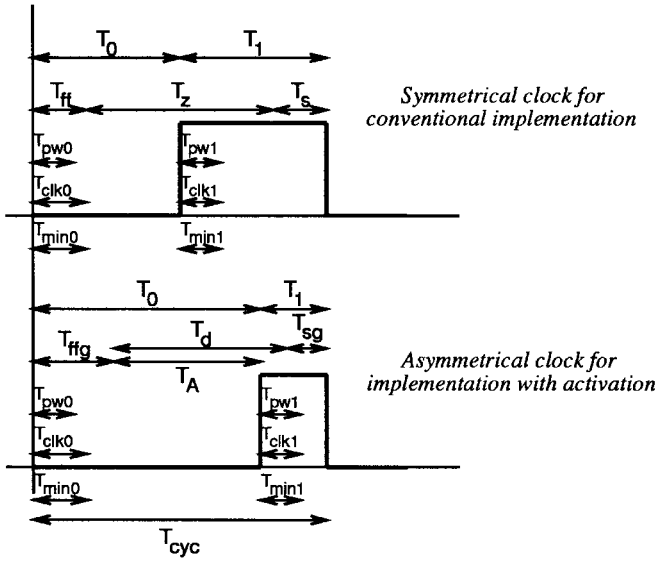


Fig. 10. Cycle for conventional case and for activation case.

2) Gated Flip-Flop

In contrast to the conventional case, for a network with ffg the cycle time is determined by the delay of both signals A and d. As discussed in Section III, signal A has to become stable during the part of the cycle in which the clock is low. That is, the low subcycle time has the restriction

$$T_0 \geq \max(T_{\rm ffg} + T_A, T_{\min 0})$$

where T_A is the delay for the logic calculating A, and $T_{\rm ffg}$ is the transmission delay of the gated flip-flop (this is somewhat larger than that of the nongated flip-flop because of the integrated AND gate). Since for the high subcycle $T_1 \geq T_{\min 1}$, we obtain

$$T_{\rm cyc} \geq \max(T_{\rm ffg} + T_A, T_{\min 0}) + T_{\min 1}.$$

Moreover, similarly to the conventional case, for signal d

$$T_{\rm cyc} \geq T_{\rm ffg} + T_d + T_{sg}$$

where T_d is the delay of the logic function for the input D of the flip-flop and T_{sg} is the new set-up time. For practical cases, $T_{\rm ffg} > T_{ff}$ and $T_s > T_{sg}$, because of the internal delay produced by the gated clock.

Combining the last two expressions we obtain

$$T_{\rm cyc} \geq \max[(T_{\rm ffg} + T_d + T_{sg}), (\max((T_{\rm ffg} + T_A), T_{\min 0}) + T_{\min 1})]$$

TABLE III
TIMING CHARACTERISTICS OF FLIP-FLOPS STRUCTURES

	T_{ff}	$T_{su}(D)$	$T_{su}(A)$
ffconv	6.0+0.2L	1.2	-
ffg	6.2+0.2L	1.2	0
ffxor	7.5+0.2L	1.2	2.3
ffor	6.7+0.2L	1.2	0.7
ffnand	6.7+0.2L	1.2	0.7

which is shown in Fig. 10. For practical cases, usually $T_{\text{ffg}} + T_A > T_{\min 0}$, and thus

$$T_{\text{cyc}} \geq T_{\text{ffg}} + \max(T_d + T_{sg}, T_A + T_{\min 1})$$

Note that when using the conventional flip-flop, the delay of signal z can be spread out over the whole cycle (low and high subcycles), whereas when using the activation signal there is a separate restriction on the low subcycle. Consequently, to obtain the minimum delay it might be necessary to have an asymmetrical clock.

For a particular situation the expressions given above have to be compared to determine the corresponding cycle times. The cycle time of the case with activation signal might be larger than that of the conventional case because

1) The delay to produce A might be larger than that to produce z;
2) The transition time of the gated flip-flop is larger than that of the conventional flip-flop;
3) It might be necessary to add $T_{\min 1}$ to the other components.

However, in some cases the delay of the signal A might be shorter than the delay of z so that $T_z - T_A$ can be used for the high part of the cycle, and still achieve the same cycle time. Moreover, the timing restriction of signal A may be relaxed or even eliminated if the logic generating A is not on the critical path. This may happen, for example, in the least-significant bits of a carry-ripple adder or when processing low-frequency signals.

In [4], a circuit to generate a gated clock is described without the timing restriction of the activation AND gate. However, this circuit is complex and, therefore, only appropriate when generating a common gated clock for several flip-flops, which is not our case.

A. Example: Redundant Accumulator

We now consider the redundant accumulator to illustrate the effect on the cycle time.[2] All times are in units corresponding to the propagation delay of a 2-input NAND gate, with unit load. The timing characteristics of the flip-flop structures we use are given in Table III. Note the two set-up times: $T_{su}(D)$ is the set-up time of flip-flop input D with respect to the trailing edge of the clock, whereas $T_{su}(A)$ is the setup of the signal applied to the input of the XOR, OR, or NAND gate with respect to the leading edge of the clock.

[2] Another example is given in [10].

The cycle time for the conventional implementation is ($T_z = 6.0$, measured)

$$T_{\text{cyc}} \geq 7.2 + 6.0 + 1.2 = 14.4$$

For the implementation with activation signal and special coding ($T_A = 2.9$)

$$T_{\text{cyc}} \geq 7.4 + 2.9 + 1.2 + T_{\min 1} = 11.5 + T_{\min 1}$$

Comparing the two values for the cycle time we can see that the implementation with activation signal is not slower than the conventional implementation for $T_{\min 1} \leq 2.9$.

VI. CONCLUSIONS

Synchronous systems require the clock that dictates the temporal behavior of the system. A significant amount of energy is wasted to conservatively ensure a proper synchronization among different components.

In this paper, fine-grain clock activation techniques have been proposed to reduce the waste of energy in flip-flops with low activity. In some applications, such as sequential multipliers and accumulators, energy savings of about 25% (for the multiplier) and 50% (for the accumulator) have been obtained by combining the clock activation with special coding approaches for redundant data representations. The savings are even more significant for specific applications such as the sampling of slow-varying signals.

Further effort is required to investigate efficient implementations of flip-flops. On the other hand, techniques for automatically selecting the most appropriate flip-flop structure according to activity criteria must also be explored.

REFERENCES

[1] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-based sequential logic optimization for low power," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 426–436, Dec. 1994.
[2] R. I. Bahar, H. Cho, G. D. Hachtel, E. Macii, and F. Somenzi, "A symbolic method to reduce power consumption of circuits containing false paths," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1994, pp. 368–371.
[3] L. Benini and G. De Micheli, "Transformation and synthesis of FSM's for low-power gated-clock implementation," in *Int. Symp. Low Power Design*, Apr. 1995, pp. 21–26.
[4] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*. Boston, MA: Kluwer Academic, 1995.
[5] M. D. Ercegovac and T. Lang, "Low-power accumulator (correlator), in *Proc. Int. Symp. Low Power Electron.*, Oct. 1995, pp. 30–31.
[6] S. Gary, P. Ippolito, G. Gerosa, C. Dietz, J. Eno, and H. Sánchez, "PowerPC 603™, a microprocessor for portable computers," *IEEE Design Test of Comput.*, pp. 14–23, Winter 1994.
[7] P. Groeneveld and P. Stravens, "Ocean: The Sea-of-Gates design system," Tech. Rep., Delft Univ. Tech., 1993.
[8] R. Hossain, L. D. Wronski, and A. Albicki, "Low power design using double edge triggered flip-flops," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 261–265, June 1994.
[9] U. Ko, P. T. Balsara, and W. Lee, "A self-timed method to minimize spurious transitions in low power CMOS circuits," in *Proc. Int. Symp. Low Power Electron.*, Oct. 1994, pp. 62–63.
[10] T. Lang, E. Musoll, and J. Cortadella, "Individual flip-flops with gated clocks for low-power datapaths," Tech. Rep., UPC-DAC-1996-26, Dept. Comput. Architect., Polytech. Univ. Catalonia, 1996, ftp://ftp.ac.upc.es/pub/reports/DAC/1996/UPC-DAC-1996-26.ps.Z
[11] LSI LOGIC. *LCA500K Preliminary Design Manual*, Nov. 1994.

[12] T. H. Meng, B. M. Gordon, E. K. Tsern, and A. C. Hung, "Portable video-on-demand in wireless communication," *Proc. IEEE*, vol. 83, pp. 659–680, Apr. 1995.
[13] S. Sheng, A. P. Chandrakasan, and R. W. Brodersen, "A portable multimedia terminal," *IEEE Commun. Mag.*, pp. 64–75, Dec. 1992.
[14] A. J. van Gerenden, "SLS: An efficient switch-level timing simulator using min-max voltage waveforms," in *Proc. VLSI 89 Conf.*, Aug. 1989, pp. 79–88.

Tomas Lang received the electrical engineering degree from the Universidad de Chile in 1965, the M.S. degree from the University of California, Berkeley, in 1966, and the Ph.D. degree from Stanford University, Stanford, CA, in 1974.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of California, Irvine. Previously, he was a Professor with the Computer Architecture Department, Polytechnic University of Catalonia, Spain, and a faculty member of the Computer Science Department, University of California, Los Angeles. His primary research and teaching interests are in digital design and computer architecture with current emphasis on high-speed and low power numerical processors and multiprocessors. He is a coauthor of a textbook on digital systems, two research monographs, one IEEE Tutorial, and the author or coauthor of research contributions to scholarly publications and technical conferences.

Enric Musoll graduated in computer science from the Universitat Polytècnica de Catalunya in 1993 and received the Ph.D. degree, also in computer science, from the same University in 1996.

He joined National Semiconductor Corp. (Integrated Processor Group), Santa Clara, CA, in December 1996 as a Senior Design Engineer. His research interests include high-level synthesis techniques for low power, low power arithmetic circuits, and low power architectures.

Jordi Cortadella received the degree in computer science from the Universitat Politècnica de Catalunya in 1985 and the Ph.D. degree from the same University in 1987.

He is a Professor with the Department of Computer Architecture, Universitat Politècnica de Catalunya, Barcelona, Spain. He was a Visiting Scholar at the University of California, Berkeley, in 1988. His research interests include computer-aided design of VLSI systems, with emphasis on synthesis and verification of asynchronous circuits, low power circuits and computer arithmetic. He has coauthored more than 70 research papers in technical journals and conferences.