Appears in *19th Conference on Advanced Research in VLSI*, Salt Lake City, UT, March 2001

Activity-Sensitive Flip-Flop and Latch Selection for Reduced Energy

Seongmoo Heo, Ronny Krashinsky, and Krste Asanović
MIT Laboratory for Computer Science, Cambridge, MA 02139
{*heomoo,ronny,krste*}*@mit.edu*

Abstract

This work presents new techniques to evaluate the energy and delay of flip-flop and latch designs and shows that no single existing design performs well across the wide range of operating regimes present in complex systems. We propose the use of a selection of flip-flop and latch designs, each tuned for different activation patterns and speed requirements. We illustrate the use of our technique on a pipelined MIPS processor datapath running SPECint95 benchmarks, where we reduce total flip-flop and latch energy by over 60% without increasing cycle time.

1. Introduction

Flip-flops and latches (collectively referred to as timing elements in this paper) are critical components in modern synchronous VLSI designs. Timing element (TE) design has a large impact on both system cycle time and system energy consumption and consequently there has been significant interest in the development of fast and energy-efficient TE circuits [2, 10, 11, 12, 14, 15, 16, 17, 18]. The evaluation methodology presented in previous work often employs a very limited set of data patterns and has usually assumed that the clock switches every cycle [10, 11, 12, 14, 15, 17, 18]. In real VLSI designs, however, there is a wide variation in clock and data activity across different TE instances.

In this paper, we show that there can be significant energy savings if each TE instance is selected from a heterogeneous library of designs, each tuned to different operating regimes. For example, low-power microprocessors make extensive use of clock gating [6, 7], resulting in many TEs whose energy consumption is dominated by input data transitions rather than clocking, and for which we should select devices with low energy on data transitions. Other TEs, in contrast, have negligible data input activity but are clocked every cycle, hence for these we should select TE designs with low clock transition energy. Previous work has also focused on the delay or energy-delay product of TEs, but real designs often include many TEs that are not on the critical path. This timing slack can be exploited by using slower, lower energy TEs.

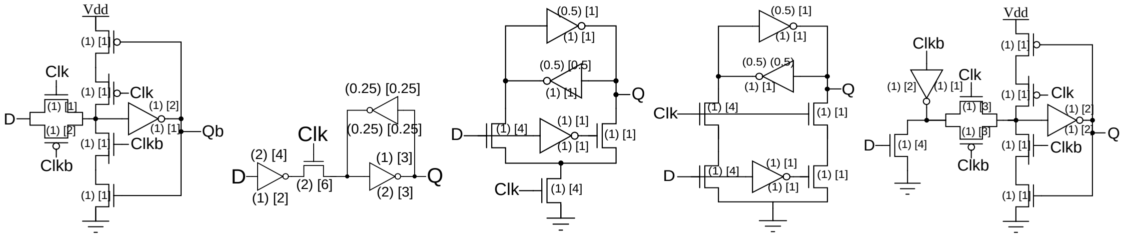
We use detailed energy analysis to compare a number of TE designs in this paper, including designs that exploit particular combinations of signal activity and timing slack. To demonstrate the potential savings from activity-sensitive TE selection, we instrument a pipelined MIPS microprocessor datapath design to gather statistics on TE activity, and simulate five SPECint95 benchmarks for a total of 2.7 billion CPU cycles. We then show that selecting appropriate TEs can reduce total TE energy without increasing cycle time.

Designing with a heterogeneous mix of flip-flop and latch structures may have the disadvantage of complicating timing verification. However, advanced designs with clock gating already perform verification for each local clock independently [1], and in this case the added complexity is minimal. Additionally, many of the alternative TE structures are used on non-critical timing paths for which verification is usually relatively straightforward.

In this work, we select flip-flop and latch structures based on activation patterns and timing slack. When selecting TE structures for a real design, more factors would come into play, including: input drive and output load, presence of differential inputs, desirability of complementary outputs, robustness to clock skew and process variations, and the ability to provide time-borrowing. These factors will tend to limit the set of designs from which TEs are selected.

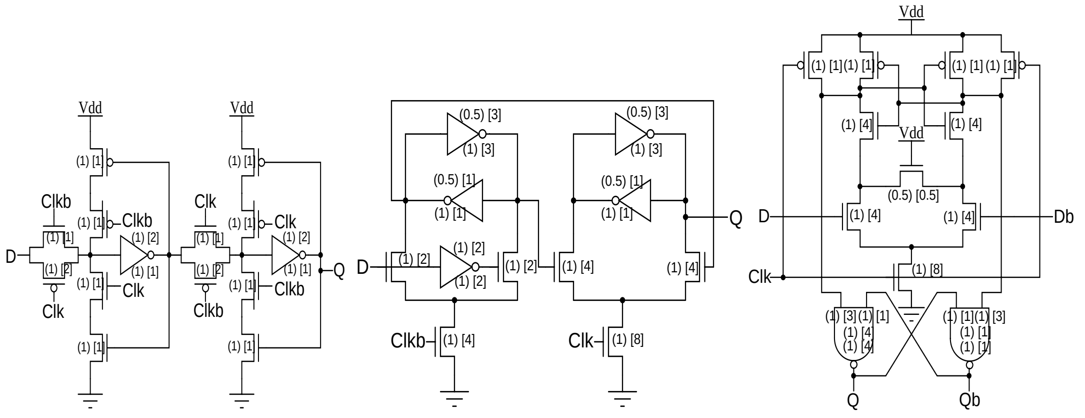
Other related work has explored the use of timing slack to reduce energy in non-critical gates: traditional transistor sizing uses smaller transistors, cluster voltage scaling [19] uses a lower supply voltage, multiple threshold voltages can be used to reduce leakage current [4, 5], or series transistors can be added to reduce leakage currents in a single threshold process [9]. These techniques are also applicable to TE design, but to our knowledge this paper is the first work that systematically exploits signal activity to reduce energy by changing the TE structure.

The paper is organized as follows. Section 2 presents a range of TE designs targeted for particular operating regimes. Section 3 describes our methodology for characterizing the energy profile of a given TE design and presents detailed simulation results for the set of candidate TE designs. Section 4 shows how the relative energy ranking of the TE designs varies widely depending on signal activity and on allowable slack. Sections 5 and 6 present results from applying activity-sensitive TE selection to a MIPS processor datapath, and Section 7 concludes.



(a) PPCLA (b) PTLA (c) SSALA (d) SSA2LA (e) CPNLA

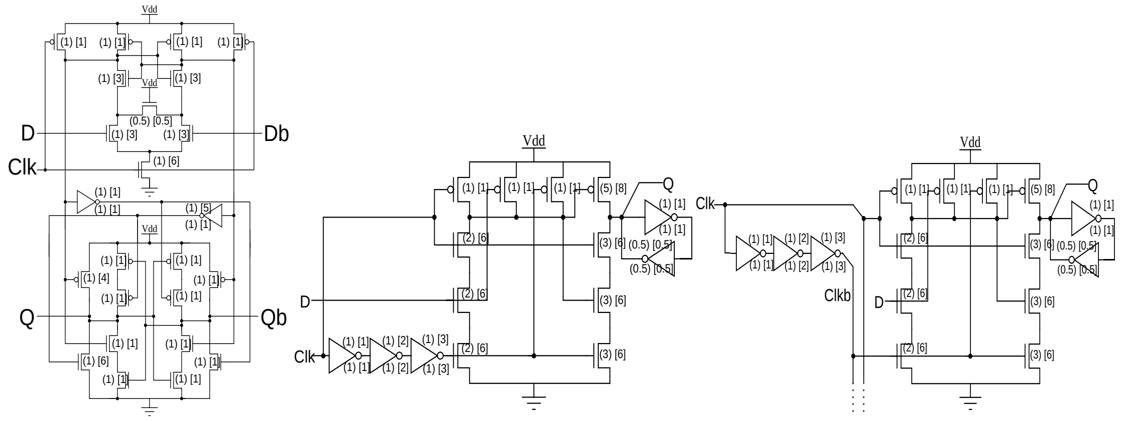
Figure 1. High-enabled latch designs. Transistor sizes are shown for a low-power design (in parentheses: (n)) and a high-speed design (in brackets: [n]). A transistor labeled with size n means that its W/L ratio is n times that of a minimum-sized transistor. For gates, the sizes of all transistors are shown.



(a) PPCFF

(b) SSAFF

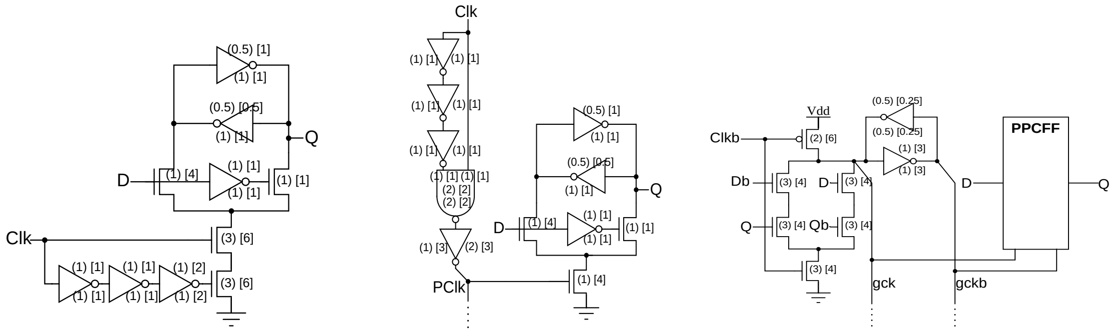
(c) SAFF



(d) MSAFF

(e) HLFF

(f) HLSFF



(g) SSAPL

(h) SSASPL

(i) CCPPCFF

Figure 2. Positive-edge-triggered flip-flop designs. Transistor sizes are labeled as in Figure 1.

2. Latch and flip-flop designs

Figures 1 and 2 present schematics for the latch and flip-flop designs we evaluated. To allow arbitrarily low clock frequencies and to allow clocks that can be gated in either phase, we restricted our designs to include only fully static structures. We used only single-rail input and output signals, and where TEs had complementary outputs we loaded only the selected output. Although not covered in this paper, we expect that our technique will also accommodate dynamic and/or complementary TEs. To ensure design robustness, we required that circuits have input buffers to isolate input sources from any actively driven feedback nodes (e.g., PTLA Figure 1(b)). We assume that both true and inverted clock signals are generated by clock buffers and so do not insert local clock inverters (although some pulsed latch designs require local inverters to generate pulses). Also, we do not penalize inverting TEs (e.g. PPCLA) because in general it is not obviously preferable to have either true or complement output. For each TE design, we developed both a low-power version and a high-speed version by sizing the transistors accordingly.

Figure 1(a), PPCLA, is a transparent latch based on the PowerPC 603 design, which is known to be reasonably fast and energy-efficient [17]. Figure 1(b), PTLA, is a pass-transistor latch, which we chose because of its low clock load. Figure 1(c), SSALA, is a latch based on a fully static differential sense amp, which we chose for its low clock load. Figure 1(d), SSA2LA, is a minor variant of SSALA, which has greater clock load but has lower data transition energy while clock is gated. Figure 1(e), CPNLA, is a PPCLA preceded by a clocked pseudo-NMOS input buffer. The pseudo-NMOS input buffer reduces the input loading of this latch and so reduces input data transition energy when the latch is closed. When the latch is transparent, the p-transistor in the clocked inverter acts as the pseudo-NMOS load and so dissipates considerable static power when the data input is high.

Figure 2(a), PPCFF, is a flip-flop design using master-slave PowerPC-style latch stages, which is known to have low energy and delay [17]. Figure 2(b), SSAFF, is a master-slave flip-flop using static sense-amp latch stages which we include for its low clock load. Figure 2(c), SAFF, is the StrongARM flip-flop [3]. Figure 2(d), MSAFF, is a StrongARM flip-flop with a modified output stage [15] that reduces output delay for higher loads.

We also measured the performance of various pulsed latch structures, which all employ an edge-triggered pulse generator to provide a short transparency window. Compared to flip-flops with master-slave latch designs, pulsed latches have the advantages of requiring only one latch stage per clock cycle and of allowing time-borrowing across cycle boundaries. The major disadvantages of pulsed latch structures are the increased susceptibility to timing hazards and the energy dissipation of the local clock pulse generators. The clock pulse generators can be shared among a few latch cells to reduce energy, although care must be taken that the pulse shape does not degrade due to wire delay, signal coupling and noise. We measured designs both with individual pulse generators and with pulse generators shared among four latch bits, in which case we divide the energy used by the pulse generator among the four latch instances.

Figure 2(e), HLFF, is the hybrid latch flip-flop [2] which operates as a pulsed transparent latch design and which is generally regarded as one of the fastest known flip-flop designs.

Figure 2(f), HLSFF, is the hybrid latch flip-flop with a shared inverter chain. Figure 2(g), SSAPL, is a pulsed version of SSALA with an individual pulse generator circuit while Figure 2(h), SSASPL, is the same structure but with a shared pulse generator. Note that the two series transistors in SSAPL are replaced by a single transistor in SSASPL.

Finally, Figure 2(i), CCPPCFF, is a conditional clocking flip-flop based on the design presented in [18], which in turn is an improvement on the designs presented in [14] and [16]. The goal of this design is to reduce energy when the input data does not change by gating the clock within the flip-flop.

3. Delay and energy characterization

Our test-bench setup is similar to [17] as shown in Figure 3. In order to have realistic input signals, the data input was driven with a minimum-sized inverter which was itself driven by a loaded minimum-sized inverter. The clock inputs were designed to simulate a local clock buffer, and the clock drivers were sized to give equal clock rise and fall times for each TE design. The TE outputs were loaded with a 7.2 fF capacitance, simulating a fanout of four minimum-sized inverters (FO4-min). Other studies [12, 15, 17] use strong input drivers and much larger output loads (200 fF). However, we have extracted capacitance values for a processor datapath (described below) including transistor gate and drain capacitances and wire substrate and coupling capacitances; and we found that over 40% of TEs have output loads less than the FO4-min load, over 60% have loads less than twice this amount, and none have loads greater than 60 fF. For brevity, we here consider only one size of output load but in general TE characterization should consider a variety of loads; we are investigating TE load sensitivity in ongoing work.

The TE designs were implemented in a 0.25 μm TSMC CMOS technology. Layouts were extracted using the SPACE 2D extractor [20] which extracts layout parasitics including capacitance to substrate, fringe capacitance, crossover coupling capacitance, and capacitance between parallel wires. All tests were run under nominal conditions of Vdd=2.5 V and T=25 ^{o}C.

Figure 4 shows the delays for both versions of each timing element (low-power and high-speed). For latches, delay is defined as the D-Q propagation delay. For flip-flops, we used the methodology proposed by [17] in which delay is defined as the minimum D-Q delay (in general the C-Q delay changes depending on when D arrives in relation to C, and there is some optimal arrival time that minimizes the total D-Q delay). These delays were obtained using HSpice.

We rely on accurate energy models to characterize candidate flip-flop and latch designs. Traditionally, the power consumption of flip-flop and latch designs has been measured using an un-gated clock and a small number of input activation patterns [10, 11, 12, 14, 15, 17, 18]. Instead, we adopt a more accurate methodology based on [21] in which all possible states of the TE are enumerated and the energy consumption of each state transition is measured. Canonical state transition diagrams for latch and flip-flop designs are shown in Figure 5.

In general, the state transition diagram for a given flip-flop or latch design may be more intricate than these canonical examples because the design may have internal nodes which are not uniquely determined by the values of C, D, and Q. In this case, the design has two or more distinct states for a given CDQ combination; its internal nodes have different values depending on the sequence of transitions taken to obtain those C, D, and Q values [21].

To characterize the TE designs, we simulated each transition using HSpice, and measured the energy consumption. The output energy of the shaded inverters in Figure 3 was included (as in [17]), but the energy dissipated on the output load capacitance was not (the purpose of this capacitor is only to simulate reasonable output signal slopes). The resulting energy numbers for our TE designs are shown in Table 1 and Table 2. When flip-flops or latches have two states corresponding to some CDQ combination, both energy numbers are shown for transitions leaving these states. We note that these differences are usually small, and for the remainder of this paper we use the average value for each transition to simplify the analysis.

Since the CPNLA design has static current dissipation when C and D are both high, we must make some assumptions in order to characterize its energy usage. We assume that the clock is gated low, so that the clock input never remains high for more than half a clock period, and we assume that the clock cycle time is a pessimistic 32 FO4 delays. Thus, in Table 2, whenever there is a transition into a state where C and D are both high, we include in the energy value the static current energy consumed during half a clock period. If D goes low during this time, the static current path will be broken, but we always assume worst case timing so that the static current lasts for the full half cycle.

4. Energy analysis

In order to more easily analyze the energy numbers in Tables 1 and 2, we constructed several example waveforms shown in Figure 6. These tests are designed to exemplify the different operating regimes for flip-flops and latches. For example, Tests 1 and 2 emphasize clock activity, while Tests 3 and 4 emphasize data activity. Tests 5, 6, and 7 exhibit high clock, input data, and output data activity. Test 8 has both clock and input data activity, but no output activity.

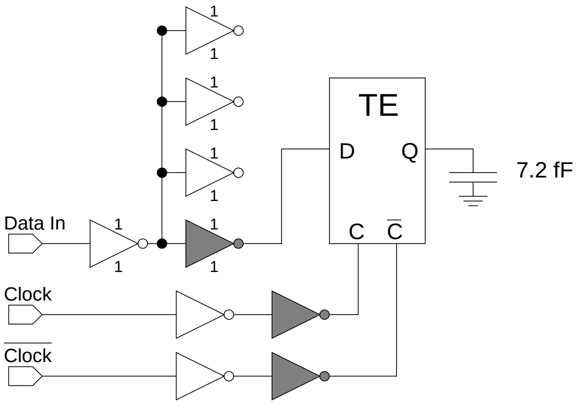


Figure 3. TE test bench.

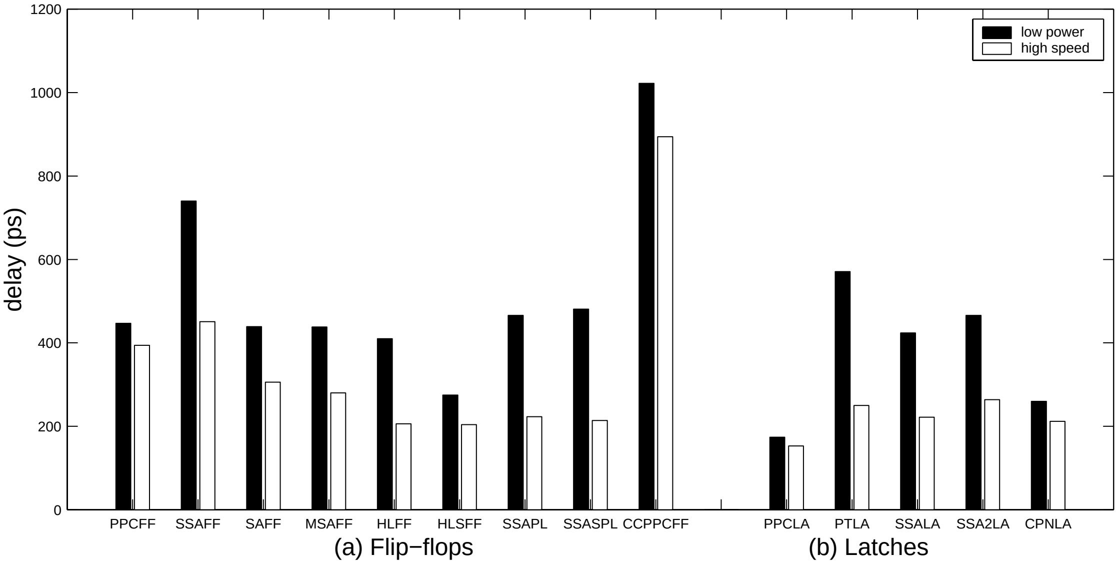
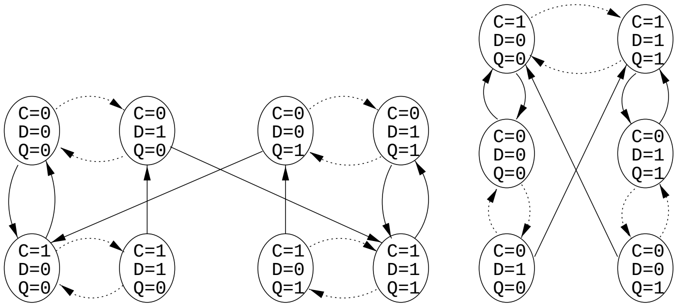


Figure 4. Delay for flip-flops and latches.



(a) Flip-flop (b) Latch

Figure 5. Canonical state transition diagrams for a positive-edge-triggered flip-flop (a) and a high-enabled latch (b). States are based on the clock input (C), data input (D), and data output (Q) levels, and transitions are based on changes in D (dotted arrows) or C (solid arrows).

	000 ↓ 100	001 ↓ 100	010 ↓ 111	011 ↓ 111	100 ↓ 000	110 ↓ 010	101 ↓ 001	111 ↓ 011	000 ↓ 010	100 ↓ 110	101 ↓ 111	001 ↓ 011	010 ↓ 000	110 ↓ 100	111 ↓ 101	011 ↓ 001
Low-Power Flip-Flop																
PPCFF	48.4	95.5 95.4	89.2 89.0	47.6	46.3 46.0	100.9	91.5	49.1 46.8	68.1	19.4 19.2	19.4	68.1 68.0	49.7 49.7	6.9	6.9 6.9	51.2
SSAFF	21.1	92.2	103.8	21.2	21.9	101.8	101.0	21.9	115.9	56.1	43.2	114.2	103.1	33.4	37.4	103.7
SAFF	65.8	112.9	118.0	68.1	53.9	54.2	59.8	61.9	26.4	28.3	28.2	26.5	15.6	17.0	17.8	15.6
MSAFF	96.2	156.2	149.8	98.7	93.0 95.7	98.5 91.7	87.3 90.9	94.0 88.3	26.5	28.3 28.3	28.2 28.2	26.6	15.9	16.9 17.0	17.8 16.9	15.7
HLFF	106.4 129.3	188.8 183.3	330.3	237.2	91.4 92.4	102.3	113.1	123.5	24.5 24.5	18.2 15.4	15.6	24.7 22.6	6.0	10.2	10.5	6.0
HLSFF	49.7 71.8	138.6 132.3	273.6	207.1	66.1 66.0	76.5	84.7	95.5	27.9 35.7	18.1 16.1	16.5	27.6 23.4	9.3	10.1	10.3	9.3
SSAPL	98.4	187.2	181.9	99.3	64.8	74.6	72.9	65.8	72.7	82.2	70.1	53.1	39.7	53.6	52.0	47.6
SSASPL	68.8	140.7	151.9	68.8	19.5	19.5	19.5	19.5	49.8	49.8	37.0	37.0	27.4	27.4	30.3	30.3
CCPPCFF	21.4	416.9 416.7	366.9 366.8	21.5	27.6 43.6	268.4	276.8	43.4 27.5	278.4	71.3 84.9	61.6	138.3 149.0	96.8 102.6	39.8	63.7 54.3	248.6
High-Speed Flip-Flop																
PPCFF	57.9	115.3 115.1	97.8 98.0	49.3	47.1 47.0	119.5	106.6	57.7 54.9	87.7	19.6 19.5	19.9	88.4 88.3	61.5 61.9	9.3	9.2 9.1	62.1
SSAFF	66.5	273.8	185.4	66.9	41.4	199.8	196.2	41.0	216.5	92.5	71.5	205.9	180.1	55.4	60.3	191.5
SAFF	164.8	246.9	257.2	164.7	105.1	97.7	110.4	125.4	39.8	48.6	48.6	41.9	29.6	35.6	36.2	26.9
MSAFF	211.4	288.5	263.8	172.9	169.1 173.0	172.8 168.1	125.7 129.5	134.5 130.4	35.6	43.2 43.1	42.5 42.5	36.4	26.8	28.1 28.2	29.1 28.9	24.0
HLFF	174.7 209.3	272.3 260.3	443.6	382.4	175.5 179.8	212.7	217.8	251.9	51.5 51.2	29.7 24.3	24.7	50.8 45.9	5.6	16.0	15.1	5.5
HLSFF	89.3 125.9	210.4 196.3	397.6	325.6	167.0 166.2	194.0	206.4	233.2	51.8 59.2	29.3 27.2	26.8	51.7 46.1	5.8	16.8	15.5	5.8
SSAPL	135.3	254.9	223.6	136.1	94.3	110.8	110.5	96.8	100.7	130.8	108.9	80.4	43.4	73.1	77.1	65.7
SSASPL	108.6	234.7	209.4	108.5	19.5	19.5	19.5	19.5	101.2	101.2	68.7	68.7	39.7	39.7	60.3	60.3
CCPPCFF	44.7	414.1 414.1	383.6 383.1	45.4	36.9 59.0	342.3	335.1	59.2 36.6	340.0	64.9 97.5	68.5	170.1 173.6	116.3 121.6	48.1	77.4 44.9	296.7

Table 1. Flip-flop energy consumption. The energy is shown in fJ for each state transition corresponding to Figure 5(a) (the states shown refer to CDQ values). Two energy numbers are given if the design actually has two internal states which correspond to the initial CDQ state of a transition.

	000 ↓ 100	001 ↓ 100	010 ↓ 111	011 ↓ 111	100 ↓ 000	111 ↓ 011	000 ↓ 010	001 ↓ 011	010 ↓ 000	011 ↓ 001	100 ↓ 111	111 ↓ 100
Low-Power Latch												
PPCLA	22.8	56.5	79.8	21.2	23.4 24.4	24.9 24.7	19.2	18.0	6.1	6.8	77.1 73.5	48.2 47.0
PTLA	18.3	226.5	95.0	29.3	0	0	32.3	32.4	32.0	30.1	90.8	266.8
SSALA	21.9	93.8	105.0	21.9	0	0	49.8	37.0	27.4	30.3	110.4	91.2
SSA2LA	23.9 27.0	98.9	107.3	26.1 23.9	0	0	33.5 32.9	32.9	23.7	24.4 23.7	119.2	99.7
CPNLA	45.0	74.4	1051.8	897.9	45.2 46.7	71.1 71.1	16.9	16.9	1.5	1.6	1100.5 1047.6	128.4 128.3
High-Speed Latch												
PPCLA	22.7	54.5	71.8	24.6	25.9 27.1	24.3 24.6	19.7	18.0	8.2	9.1	68.0 68.4	45.1 44.8
PTLA	24.7	152.4	141.7	54.4	0	0	54.4	55.3	67.1	59.9	156.8	188.1
SSALA	47.4	173.5	148.2	47.3	0	0	101.2	68.7	39.7	60.3	135.8	145.8
SSA2LA	30.0 35.8	188.1	120.8	47.3 42.1	0	0	55.4 51.6	51.8	27.3	30.4 28.4	153.1	171.0
CPNLA	78.2	115.2	1873.9	1620.0	65.0 66.6	114.0 113.9	34.9	34.9	0	0	1965.5 1868.1	219.6 222.0

Table 2. Same as Table 1 for latches.

For each test, we used Tables 1 and 2 to calculate energy. The resulting energy consumption is shown in Table 3. We can see that the optimal flip-flop or latch for each regime varies considerably; some designs perform extremely well in certain regimes, but extremely poorly in others. For example, in Test 2 the low power SSAFF design uses 8 times *less* energy than the HLFF structure, but in Test 3 it uses 7 times *more* energy. Another good example of a TE specialized for an operating regime is CPNLA; this latch design is by far the best choice for Test 3, but by far the worst choice in all other cases.

In these results we also see the flaw in the methodology of many flip-flop and latch analyses which test only a limited set of data activations with clock always un-gated [10, 11, 12, 14, 15, 17, 18]. These studies typically look at Tests 5, 6, and 7; however, we see that the optimal TE choice may be very different if we take Tests 1-4 into consideration. Also, in these studies, the TEs are typically optimized for energy-delay product. Our results show that if we size a design for high-speed and low-power separately, the energy usage can differ substantially. When the TE is not on a critical path the low-power design should be used, and when timing is critical the high-speed design should be used. If TEs are only optimized for energy-delay product, the result will be a slower circuit that burns more power.

Another important observation is that CCPPCFF never uses less energy than SSAFF, even when data is inactive. This is because both designs have two transistor gate loads on the clock. Additionally SSAFF is significantly faster and less complex than CCPPCFF, so we conclude that it is always a better choice. The analyses in [14, 16, 18] which advocate an individually gated clock are unfair in that they only compare their designs with flip-flops that have eight transistor gate loads on the clock.

5. Processor design and simulation

To evaluate the effectiveness of designing with diverse flip-flop and latch structures, we tested our idea on a processor datapath. Our processor design is a classic 32-bit MIPS RISC five-stage pipeline (R3000 compatible), including caches and system coprocessor registers. We are implementing this design as part of a low-power processor project. To

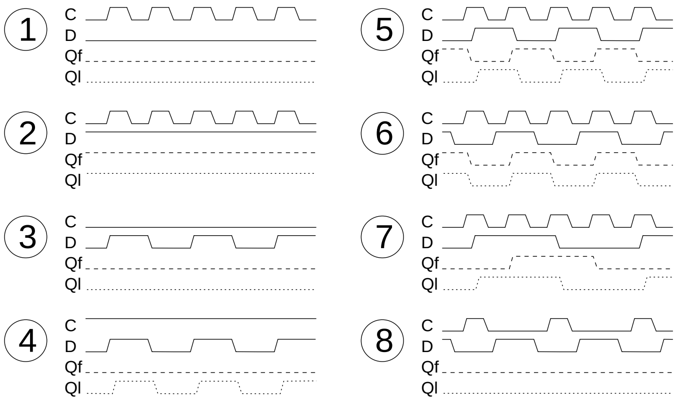


Figure 6. Waveforms for flip-flop and latch tests. The data output waveforms are shown for a positive-edge-triggered flip-flop (Qf, dashed), and a high-enabled latch (Ql, dotted).

Test:	1	2	3	4	5	6	7	8
Low-Power Flip-Flop								
PPCFF	95	97	59	**13**	202	200	145	106
SSAFF	**43**	**43**	110	45	246	230	**133**	131
SAFF	120	130	21	23	**196**	**194**	154	**81**
MSAFF	191	190	21	23	268	267	223	117
HLFF	210	361	**15**	14	380	381	329	120
HLSFF	127	303	21	14	299	306	253	84
SSAPL	163	165	56	68	325	310	228	138
SSASPL	88	88	39	39	206	206	137	83
CCPPCFF	57	57	189	59	733	691	378	218
High-Speed Flip-Flop								
PPCFF	105	106	75	14	234	233	166	127
SSAFF	108	108	198	74	504	475	287	252
SAFF	270	290	35	42	399	401	329	170
MSAFF	383	305	31	36	461	458	394	222
HLFF	370	634	29	22	591	598	541	213
HLSFF	274	559	31	23	523	531	464	168
SSAPL	230	233	72	102	454	418	317	187
SSASPL	128	128	70	70	322	322	205	135
CCPPCFF	82	105	228	57	809	765	433	269
Low-Power Latch								
PPCLA	47	46	13	**61**	**108**	**106**	77	**36**
PTLA	**18**	29	32	179	203	192	113	41
SSALA	22	**22**	39	101	123	139	**72**	50
SSA2LA	26	25	28	109	135	132	80	41
CPNLA	91	969	**9**	601	1131	631	831	55
High-Speed Latch								
PPCLA	49	49	14	57	106	103	77	39
PTLA	25	54	61	172	212	204	126	73
SSALA	47	47	70	141	188	242	118	94
SSA2LA	33	45	40	162	201	196	120	57
CPNLA	144	1734	17	1069	2008	1102	1473	89

Table 3. TE energy consumption for tests of Figure 6. The energy numbers given are in fJ per clock cycle. For the low-power TE designs, the minimum energy for each test is shown in bold.

Name	critical timing?	Description
Flip-flops		
f_recovpc	no	previous pc, used for not-taken branch recovery and link instructions
d_inst	yes	instruction, from instruction cache
d_epc	no	pc chain – for data cache miss recovery and exceptions
x_epc	no	pc chain – for data cache miss recovery and exceptions
m_epc	no	pc chain – for data cache miss recovery and exceptions
x_sd	no	store data register, before alignment
x_addr	yes	address register, sent to data cache
m_exe	no	output of execute stage for register file writeback
cp0_count	no	system coprocessor count register
cp0_comp	no	system coprocessor compare register
cp0_baddr	no	system coprocessor bad virtual address register
cp0_epc	no	system coprocessor exception program counter register
Latches		
p_pc	yes	program counter, sent to instruction cache
f_pc	no	program counter
d_rsalu	yes	register rs input to alu
d_rtalu	yes	register rt input to alu
d_rsshmd	no	register rs input to shifter and mult/div unit
d_rtshmd	no	register rt input to shifter and mult/div unit
d_aluctrl	no	alu control
x_exe	no	output of execute stage for register file writeback
x_sdalign	yes	aligned store data, sent to data cache
w_result	yes	input to register file writeback

Table 4. Description of flip-flops and latches in the datapath. The “critical timing?” field indicates whether or not the TE is on a critical path in the circuit design.

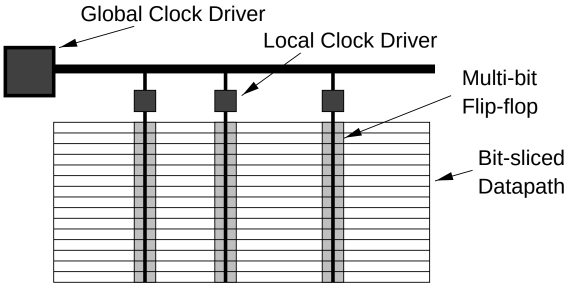


Figure 7. Datapath clocking strategy.

date, we have custom layout for the entire CPU datapath [8], and a fully functional RTL model which runs large benchmark programs using both kernel and user modes. The flip-flops and latches of our datapath are summarized in Table 4. The design contains 22 multi-bit flip-flops and latches, totaling 675 individual bits.

In the datapath design, a global clock is distributed to local clock drivers for each multi-bit (usually 32-bit) flip-flop and latch in the system (Figure 7). These drivers buffer the clock signal before sending it across the width of the datapath to trigger the individual flip-flops and latches. In these local clock drivers, we have the ability to gate the clock and effectively avoid activating the multi-bit latch or flip-flop. The processor design employs aggressive clock gating to avoid clocking flip-flops and latches whenever possible. This saves energy by eliminating the clock transitions for the gated flip-flops and latches, and also stops spurious values from propagating down the pipeline and consuming energy in downstream functional units.

In order to characterize the behavior of the flip-flops and latches in the CPU datapath, we simulated the design using a fast cycle-accurate simulator. We augmented the framework previously presented in [13] to count the relevant TE state transitions. This simulation framework tracks the input and output values of all blocks in the designs (flip-flops, adders, muxes, etc.), and is cycle-accurate for both the high and low regions of the clock period. However, it does not accurately track the timing of signals and it does not model glitches. If modeled accurately, glitching activity would have the effect of increasing the input data activity for TEs, and could possibly affect the optimal design choice. In low-power datapath designs, however, glitching activity is usually kept to a minimum.

As a test set, we chose five programs from the SPECint95 benchmarks: perl(test, primes), ijpeg(test), m88ksim(test), go(20,9), and lzw[1]. In total, the benchmarks executed 1.71 billion instructions in 2.69 billion cycles (CPI = 1.57). For each TE, we counted the number of relevant state transitions, subject to the constraints of a cycle-accurate simulator mentioned above. Negative-edge-triggered flip-flops and low-enabled latches were implemented as their positive/high counterparts, but with inverted clock signals.

6. Processor energy results

A simplified view of the data collected by the simulations is shown in Figure 8. Here, the TE state transition counts have been compressed into clock and input data activity. It is readily apparent that the various TEs have substantially different activation patterns. Also, we notice that data activity tends to be very low, while clock activation is generally much greater.

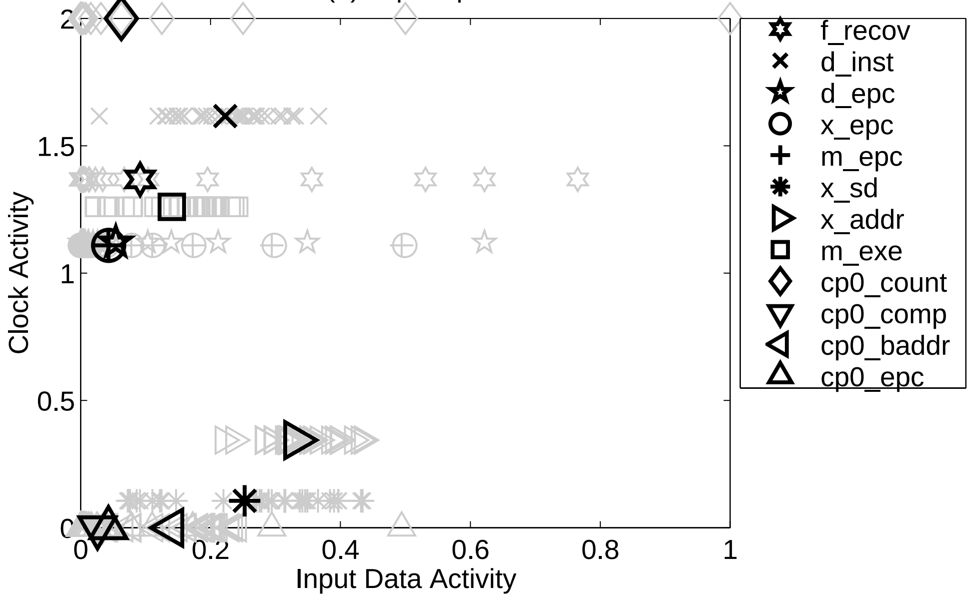
Next we show the total energy used by all TEs in the datapath if a single design is used universally. As a point of reference, the energy for the total datapath other than the flip-flops and latches (and not including caches or control logic) was about 0.21 J for these tests. Figures 9 and 10 show the TE energy plotted against the delay of each TE (from Figure 4). As long as at least one TE is on a critical path (as is the case for the CPU design), this delay has a direct impact on the maximum clock frequency of the circuit. Also plotted (for HLFF, SSASPL, and PPCLA) is the energy usage when a fast design is used for all TEs with critical timing, and the low-power version of this same design is used for non-critical TEs. This shows the improvement that would be obtained by traditional transistor sizing on non-critical timing paths.

We also show optimal points obtained using activity-sensitive selection of TE designs. One option (Lowest-Energy) is to always choose the optimal TE design to minimize the energy consumption for a particular TE in the datapath. This results in minimal energy, but the delay impact is set by the slowest TE on a critical path. The other option we show (for HLFF, SSASPL, and PPCLA) is High-Speed-Lowest-Energy (HSLE) in which a fast design is used for any timing-critical TE, and the design which results in lowest energy is used otherwise. In this study, we choose a design universally for each multi-bit TE; we found that choosing the optimal design for every bit in every TE only improved results by less than one percent. This is because the clock activity for all bits in a TE is identical, and

---

[1] This is an optimized version of the SPECint95 compress benchmark.

(a) Flip−flops



(b) Latches

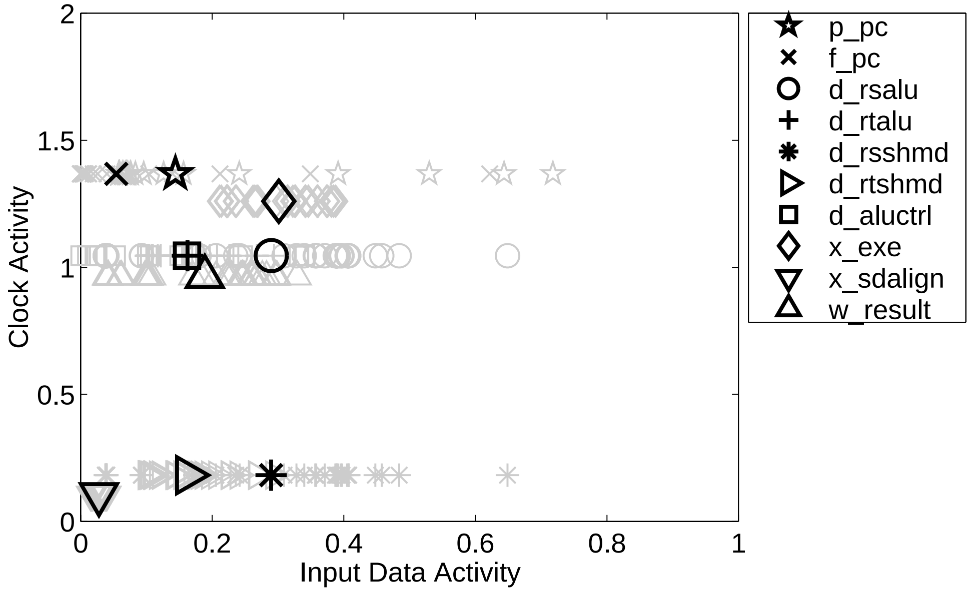


Figure 8. Clock and input data activity for flip-flops (a) and latches (b) in the CPU datapath. The activity rates given are the number of transitions per clock cycle. Note that the maximum clock activity of 2.0 indicates two transitions per cycle (rising and falling). The gray markers represent individual bits, while the black markers represent the average for each multi-bit (e.g. 32-bit) flip-flop or latch.

the data activity tends to be similar.

Table 5 shows the energy breakdown in more detail. For each TE instance, we show the energy for the fastest TE (HLFF-hs, PPCLA-hs), along with that for the lowest energy TE. We also include SSASPL-hs as a high-speed flip-flop option since it is only slightly slower than HLFF-hs (214 ps vs. 204 ps) but uses much less energy. The totals given show the energy for a fast design with homogeneous TEs, the saving achieved by transistor sizing, and the saving using HSLE activity-sensitive selection. For flip-flops, HSLE selection reduces energy by 69% compared to a fast homogeneous design using HLFF-hs, and 52% compared to a design with transistor sizing. If we start with SSASPL-hs as the base case, the saving is 43% compared to a homogeneous design, and 25% compared to a design with transistor sizing. For latches, the opportunity to save energy is reduced because they are simpler structures, and the fastest latch (PPCLA) is also quite energy efficient for the activation patterns in the datapath. Nevertheless, the energy saving with HSLE selection is 8.3% compared to a homogeneous design using PPCLA-hs, and 6.1% compared to a design using transistor sizing.

Overall, the saving we get for flip-flops and latches using HSLE activity-sensitive selection is 63% compared to a homogeneous design with HLFF-hs and PPCLA-hs, and 46% compared to a design with transistor sizing. If SSASPL-hs is used as the base case flip-flop, the HSLE saving is 35% compared to a homogeneous design, and 19% compared to a design with transistor sizing. Table 5 shows that several different TE structures are used when the processor design is optimized for both energy and speed; this validates our hypothesis that a heterogeneous mix of TE structures can result in a lower energy design without degrading performance.

7. Summary

Traditionally, designers have chosen flip-flop and latch structures to use uniformly throughout a circuit. Because of this, many studies have compared TE designs based on a limited set of activation patterns in order to determine the best universal design. The proposition of this paper is that no flip-flop or latch design is universally optimal. Designs vary significantly in parameters such as delay, clock switching energy, and input data switching energy. Two important observations allow us to use this variance to enable circuit designs with more optimal energy usage and performance. First, the activation patterns for various TEs in a given circuit may differ considerably. Second, most TEs do not lie on critical paths, and thus have ample timing slack.

Based on these observations, we propose an alternative methodology in which the designs for various flip-flops and latches are chosen from among a range of alternatives based on the local operating conditions and delay requirements. We present a variety of TE structures with separate transistor sizings for high-speed and low-power, and provide complete energy and delay characterizations. We examine several operating regimes based on clock and data activity, and find that indeed there is considerable variation in the optimal TE design for different regimes.

We apply our technique to a MIPS RISC processor design which we simulate for 2.7 billion cycles of program execution to determine flip-flop and latch activation patterns.

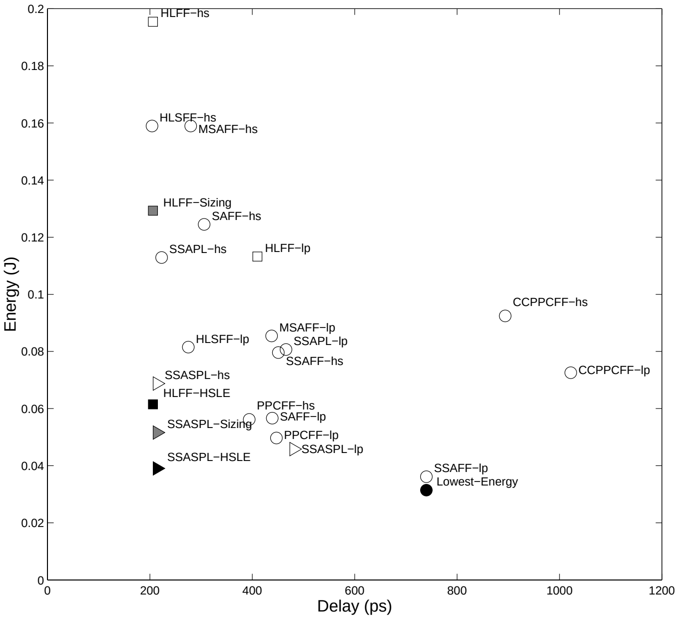


Figure 9. The total energy used by all flip-flops in the processor datapath while executing the entire benchmark test set is shown for each candidate design assuming that it is used universally. This energy is plotted against the delay of the flip-flop design, which has a direct impact on maximum clock frequency. A *-hs* suffix refers to a flip-flop design sized for high speed, while a *-lp* suffix refers to a design sized for low power. *Lowest-Energy* shows the results of using activity-sensitive selection to minimize energy for each flip-flop instance. *HLFF-Sizing* uses HLFF-hs for all timing-critical flip-flops, and HLFF-lp otherwise. *HLFF-HSLE* uses HLFF-hs for all timing-critical flip-flops, and activity-sensitive selection to pick the lowest energy design otherwise. *SSASPL-Sizing* and *SSASPL-HSLE* are analogous to the corresponding HLFF markers.

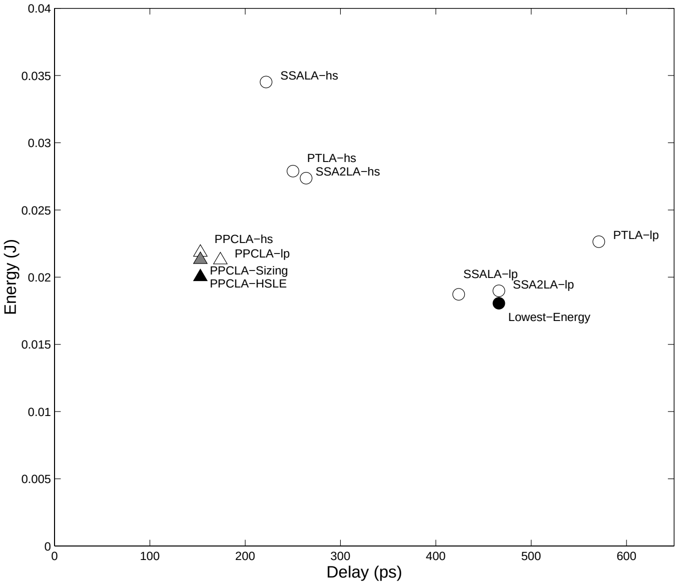


Figure 10. Same as Figure 9 for latches. CPNLA-lp and CPNLA-hs are not shown on the plot; their total energy values are 0.123 J and 0.214 J respectively.

Flip-flops				
	HLFF-hs	Lowest-Energy		SSASPL-hs
f_recovpc	25.1	SSAFF-lp	**3.57**	8.12
d_inst	**31.2**	SSAFF-lp	6.52	**12.52**
d_epc	20.5	SSAFF-lp	**2.74**	6.53
x_epc	20.3	SSAFF-lp	**2.62**	6.41
m_epc	20.2	SSAFF-lp	**2.55**	6.30
x_sd	2.6	SAFF-lp	**1.06**	2.19
x_addr	**8.0**	SAFF-lp	2.57	**4.18**
m_exe	24.6	SSAFF-lp	**4.76**	9.30
cp0_count	42.6	SSAFF-lp	**4.80**	12.07
cp0_comp	0.1	HLFF-lp	**0.03**	0.16
cp0_baddr	0.3	HLFF-lp	**0.18**	0.78
cp0_epc	0.1	HLFF-lp	**0.05**	0.23
Total	195.4		31.44	68.78
Sizing	129.3			51.62
HSLE	61.50			39.05

Latches				
	PPCLA-hs	Lowest-Energy		
p_pc	**3.22**	SSALA-lp	2.25	
f_pc	2.95	SSALA-lp	**1.72**	
d_rsalu	**3.27**	SSALA-lp	3.16	
d_rtalu	**2.81**	SSALA-lp	2.28	
d_rsshmd	0.75	PPCLA-lp	**0.70**	
d_rtshmd	0.65	PPCLA-lp	**0.63**	
d_aluctrl	1.26	SSALA-lp	**0.97**	
x_exe	3.88	SSALA-lp	**3.65**	
x_sdalign	**0.30**	SSA2LA-lp	0.27	
w_result	**2.74**	SSALA-lp	2.42	
Total	21.84		18.06	
Sizing	21.31			
HSLE	20.02			

TE total				
Total	217.2		49.5	90.62
Sizing	150.6			72.93
HSLE	81.5			59.07

Table 5. A breakdown of the total energy used by TEs in the processor datapath while executing the entire benchmark test set. Shown are energy numbers (in mJ) for the fastest TE designs (HLFF-hs, PPCLA-hs) and the designs which use the lowest energy in each instance. SSASPL-hs is also included as a high-speed flip-flop option. The total energy is shown as well as the total energy obtained using transistor sizing and the total energy using HSLE activity-sensitive selection. The bold values indicate which energy numbers are chosen with HSLE selection, based on which TEs have critical timing requirements.

We determine that, compared to a high-performance design with homogeneous flip-flop and latch structures, a processor designed with activity-sensitive selection of TE structures results in a total TE energy reduction of 63% with no loss in performance. Compared to a design which uses transistor sizing alone to reduce energy, activity-sensitive selection results in a total TE energy reduction of 46%.

8. Acknowledgments

We thank numerous helpful reviewers. This work was partly funded by an NTT graduate fellowship and by DARPA PAC/C award F30602-00-2-0562.

References

[1] D. Bailey and B. Benschneider. Clocking design and analysis for a 600 MHz Alpha microprocessor. *IEEE Journal Solid-State Circuits*, 33(11):1627–1633, November 1998.

[2] H. Partovi *et al.* Flow-through latch and edge-triggered flip-flop hybrid elements. *Digest ISSCC*, pages 138–139, February 1996.

[3] J. Montanaro *et al.* A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor. *IEEE Journal Solid-State Circuits*, 31(11):1703–1714, November 1996.

[4] T. McPherson *et al.* 760 MHz G6 S/390 microprocessor exploiting multiple Vt and copper interconnects. *Digest ISSCC*, page 96, February 2000.

[5] T. Yamashita *et al.* A 450 MHz 64-b RISC processor using multiple threshold voltage CMOS. *Digest ISSCC*, page 290, February 2000.

[6] V. Tiwari *et al.* Reducing power in high-performance microprocessors. In *DAC*, pages 732–737, June 1998.

[7] R. Gonzalez and M. Horowitz. Energy dissipation in general purpose microprocessors. *IEEE Journal Solid-State Circuits*, 31(9):1277–1284, September 1996.

[8] S. Heo. A low-power 32-bit datapath design. Master's thesis, Massachusetts Institute of Technology, August 2000.

[9] M. C. Johnson, D. Somasekhar, and K. Roy. Leakage control with efficient use of transistor stacks in single threshold CMOS. In *DAC*, pages 442–445, New Orleans, LA USA, June 1999.

[10] H. Kawaguchi and T. Sakurai. A reduced clock-swing flip-flop (RCSFF) for 63% power reduction. *IEEE Journal Solid-State Circuits*, 33(5):807–811, May 1998.

[11] U. Ko and P. Balsara. High performance, energy-efficient D flip-flop circuits. *IEEE Trans. VLSI Systems*, 8(1):94–98, February 2000.

[12] B. Kong, S. Kim, and Y. Jun. Conditional-capture flip-flop technique for statistical power reduction. *Digest ISSCC*, page 290, February 2000.

[13] R. Krashinsky, S. Heo, M. Zhang, and K. Asanović. SyCHOSys: Compiled energy-performance cycle simulation. In *Workshop Complexity-Effective Design, 27th Int. Symp. Computer Architecture*, Vancouver, Canada, June 2000.

[14] T. Lang, E. Musoli, and J. Cortadella. Individual flip-flops with gated clocks for low power datapaths. *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, 44(6):507–516, June 1997.

[15] B. Nikolić, V. Oklobdžija, V. Stojanović, W. Jia, J. Chiu, and M. Leung. Improved sense-amplifier-based flip-flop: Design and measurements. *IEEE Journal of Solid-State Circuits*, 35(6):876–884, June 2000.

[16] M. Nogawa and Y. Ohtomo. A data-transition look-ahead DFF circuit for statistical reduction in power consumption. *IEEE Journal Solid-State Circuits*, 33(5):702–706, May 1998.

[17] V. Stojanović and V. Oklobdžija. Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems. *IEEE Journal Solid-State Circuits*, 34(4):536–548, April 1999.

[18] A.G.M. Strollo, E. Napoli, and D. De Caro. New clock-gating techniques for low-power flip-flops. In *ISLPED*, pages 114–119, Rapallo, Italy, July 2000.

[19] K. Usami and M. Horowitz. Clustered voltage scaling technique for low-power design. In *Proc. Int. Symp. Low Power Electronics and Design*, pages 3–8, October 1995.

[20] N.P. van der Meijs and A.J. van Genderen. Space tutorial. Technical Report ET-NT 92.22, Technical Report, Delft University of Technology, Netherlands, 1992.

[21] V. Zyuban and P. Kogge. Application of STD to latch-power estimation. *IEEE Trans. VLSI Systems*, 7(1):111–115, March 1999.