2003년도 대한전자공학회 하계종합학술대회 논문집 제26권 제1호

저전력 RTL 설계를 위한 최적 클럭 주기 선택 알고리듬에 관한 연구

최지영, *변상준, **김희석
제천기능대학 정보통신설비과, *대덕대학 전기전자공학부, **청주대학교 전자공학과
cjy03@kopo.or.kr, *szbyun@mail.ddc.ac.kr, ** khs8392@chongju.ac.kr

A Study on Optimal Clock Period Selection Algorithm for Low Power RTL Design

Ji-young Choi, *Sang-zoon Byun, **Hi-seok Kim
Dept. of Information & Communication, Jecheon Polytechnic College
*Division of Electronic Engineering Taedok College, **Dept. of Electronic Engineering, Chongju University
cjy03@kopo.or.kr, *szbyun@mail.ddc.ac.kr, ** khs8392@chongju.ac.kr

요 약

We proposed a study on optimal clock period selection algorithm for low power RTL design.

The proposed algorithm use the way of maintaining the throughput by reducing supply voltage after improve the system performance in order to minimize the power consumption. In this paper, it select the low power to use pipeline in the transformation of architecture.

Also, the algorithm is important the clock period selection in order to maximize the resource sharing. however, it execute the optimal clock period selection algorithm.

1. 서 론

[1]VLSI 제조 및 설계 기술의 발달과 최근 휴대용 컴퓨터와 통신 시스템의 수요가 급증함에 따라 회로의 발열 문제와 전력 소모 개선하기 위하여 침의 고성능, 고집적화 경향과 저전력 회로의 설계가 점차 중요한 설계 사양으로 등장하게 되었다. 이에 따라 저전력 회로 설계자동화에 관한 연구가 활발하게 진행되고 있으며 상위 수준 합성기, 전역 소모 예측 시스템 등이 개발되어 보고되고 있다.[1,2]

일반적으로 CMOS 회로에서의 전형적인 전력 소모의 요인으로 스위칭 활동(switching activity), 누설전류(leakage current), 폐회로 전류(short-circuit current)등에 의하며, 이들 중 스위칭 동작에 의한 전력 소모가 약 90% 이상으로 가장 큰 비중을 차지한다.[3-5]. CMOS 회로에서는 데이터의 스위칭 동작이 발생하지 않을 경우 전력 손실이 없으므로, 저 전력 회로 설계에 있어서 최소의 스위칭 동작을 허용하는 것이 중요한 관건으로 적용된다. 현재 저 전력 설계를 위해 공급 전압의 감소, 스위칭 동작의 최소화 등을 통한 여러 가지 설계 방식을 제안하고 있다. 앞에서도 언급했듯이, 전력이 공급 전압의 제곱에 비례하는 관계로, 공급 전압의 감소는 큰 전력 감소를 초래할 수 있다. 하지만, 이때 공급 전압의 감소로 인해 지연 시간은 증가한다. 이런 이유로 인해 전력 감소를 위해 여러 변환 기법을 이용하여 회로의 성능을 높인 후, 원래의 성능 제한 조건을 위반하지 않는 범위 한도 내에서 전압을 낮춘다. 또한 과거에는 VLSI 설계에서 주 고려사항은 성능 및 비용 신뢰성 면적이었다. 반면 오늘날에는 무선통신 시스템, 음성 및 비디오를 기초로 한 멀티미디어 제품 휴대용 데스크탑과 같은 개인용 컴퓨터의 성장은 휴대용을 요구한다. 모든 휴대용 장치들은 고속의 계산과 복잡한 기능뿐만 아니라 저 전력 소비를 요구한다. 결론적으로, 전력의 고려는 오늘날 VLSI 설계에서 지배적인 것으로 되고 있다. 디지털 CMOS VLSI의 동적 전력 소모는 다음과 같은 식으로 표현된다.

$$P_D = \sum T_i C_i V_{dd}^2 f_s \quad \text{.....식(1)}$$

여기에서 T_i 는 평균 천이 동작수(transition activity), C_i 는 하드웨어 유닛 i 에 대한 스위칭 정전용량, V_{dd} 는 공급 전압, 그리고 f_s 는 동작 주파수를 말한다. 그러한 전력 소모에 영향을 주는

1) 본 연구는 과학기술부 · 한국과학재단 지정 청주대학교 정보통신연구센터의 지원에 의한 것입니다.

요소들을 최소화하기 위하여 전력 소비를 줄이는 위한 노력은 설계의 다양한 계층에서 고려되고 있다. 논리 회로에는 패스 로직(pass logic), 멀티 스레숄드 로직(*multi-threshold logic*), 그리고 adiabatic logic circuit[6] 등이 사용되고 LSI 아키텍쳐 에서는 병렬(parallel), 파이프라인(pipeline), 스위칭 캐패시터 감소(switching capacitance reduction), algorithm 변환 그리고 파워 관리(power management) 등의 방법 등이 적용되고 있다. 또한 compilation, 스케쥴링, 자원 할당을 포함하는 상위 레벨 합성도 전력 감소에 크게 기여하고 있다. 이런 여러 가지 방법 중 아키텍쳐 변환에 의한 전압 scaling 방식은 전력감축에 큰 효과를 나타낸다. 본 논문의 구성은 서론에 이어 2장에서는 저전력 RTL 설계를 위한 최적 클럭 주기 선택 알고리듬을 기술하고, 3장에서는 2장의 알고리듬을 토대로 다양한 벤치 마크를 통한 알고리즘의 효율성을 보이며. 마지막으로 결론을 나태낸다.

2. 저전력 RTL 설계를 위한 최적 클럭 주기 선택 알고리듬

본 논문에서는 RTL 설계를 위한 클럭 주기 선택 알고리듬을 제안한다. 먼저 소비 전력을 최소화하는 방법은 시스템의 성능을 증가시킨 다음 전압을 낮추어 처리량을 유지시키는 방법으로 본 논문에서는 파이프라인을 이용해서 성능을 최대한으로 한 뒤 소비 전력을 줄이는 방법을 선택한다. 또한 자원 공유를 최대로 하기 위해 클럭 주기 선택이 중요하다.

CMOS 게이트의 지연시간 T_d 는 다음과 같이 주어진다.

$$T_d = \frac{C_L V_{dd}}{I} = \frac{C_L V_{dd}}{k(w/L)(V_{dd} - V_t)^2} \quad \text{.....식(2)}$$

여기에서 C_L 은 게이트 정전용량, I는 출력전류, V_T 는 threshold voltage, k는 공정의존 변수, W와 L은 트랜지스터의 폭과 길이를 말한다. 즉 V_{dd} 가 V_T 에 접근함에 따라 T_d 가 증가하는 것을 알 수 있다. 고성능의 시스템 설계 시 성능 및 산출량을 보존하면서 전력량을 줄이는 것이 중요하므로 전압 축소에 따른 지연은 적절한 아키텍쳐 변형을 통하여 보상하여야 한다. 그 방법에는 크게 병렬 프로세싱과 파이프라인 두가 방법이 있다. 이중 본 논문에서는 파이프라인 기법을 채택하였다. 파이프라인 기법은 임계 경로를 래치를 분리하여 분리된 각각의 블럭을 동시에 수행하는 기법으로 임계 경로 지연이 반으로 줄어들면 산출량은 대략 두 배로 증가하게 된다. 그때 임계 경로 지연의 감소로 인하여 전압은 식(2)에 의하여 2.9V로 줄어든다. 그러므로 소비 전력을 줄이는데 큰 역할을 한다. 그러므로 최적의 선택 알고리듬을 수행한다.

2.1 문제 정의 및 가정

먼저, 다음과 같은 가정을 따른다. 첫째, 각 파이프 단계 PSi 에 표현되는 DFGi 의 파이프라인 n 단계 PS1 .....PS*n* 이고 둘째, 컴포넌트 라이브러리이며 셋째, 파이프 단계의 지연의 제약조건인 PSDelay 이고, 마지막으로 모든 클럭 주기에 따르는 범위 [clkmin, clkmax], 스케쥴된 DFG*i*와 PSDelay/clk ,지연상태의 clk 이 정의되어야 한다. 그림 1은 정의에 따른 입력 및 출력의 가정 예시를 나타낸다.

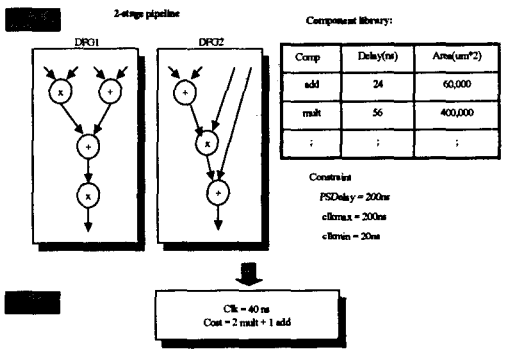

그림 1. 입출력 가정의 예

제안된 클럭 주기 선택 알고리듬은 크게 3단계로 이루어진다. 첫째, 파이프 단계 모형을 일반화하는 단계로써 클럭 주기 및 파이프 단계를 생산한다. 둘째, 후보 클럭 즉, 클럭 주기의 집합을 나타낸다. 이 과정에서는 제약조건인 파이프 단계의 지연인 PSDelay 와 파이프 단계의 모형을 만족하는 모든 지연을 얻을 수 있고, 마지막 단계에서는 자원의 예측으로써 후보 클럭에 의해 요구되는 자원의 양을 평가할 수 있는 과정이다.

2.2 파이프 단계 설계 모형

이 절에서는 파이프 단계의 설계 모형을 나타내는 절로 클럭 주기 산출을 목적으로 수행한다. 그림 2는 클럭 예측 설계 모델을 나타낸다. 이 모델의 데이터 경로는 레지스터와 기능연산자, 3상태 드라이버로 구

성된다. 2단계 버스의 구조는 레지스터와 기능연산자를 통한 연결구조로 가정한다. 또한 레지스터는 파이프 단계사이의 파이프 래치를 이용하거나 같은 파이프 단계의 다른 상태를 이용하여 임시값을 저장한다. 또한 이 모델에서는 체이닝과 멀티사이클링이 지원된다.

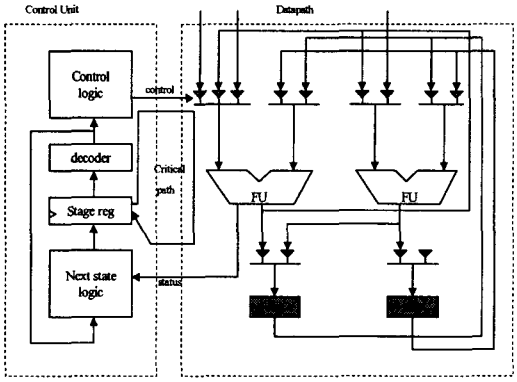

그림 2. 클럭 예측 설계 모델

이 모델에서 클럭 계산은 식(3)에 의해서 구하여진다. 단 배선 지연은 상위 레벨 합성 단계에서는 제외되었다.

$$clk = T_{PSR} + T_{DEC} + T_{CL} + T_{TR} + T_{FU} + T_{NS} + T_{sSR} + T_{Wire} \quad \text{.....식(3)}$$

T_{PSR}, T_{sSR} : 전파지연과 상태 레지스터의 셋업 시간
T_{DEC} : 디코더의 지연
T_{CL} : 컨트롤 로직의 지연
T_{TR} : 3 상태 드라이버 지연
T_{FU} : 기능연산자 지연
T_{NS} : 다음 상태 로직의 지연
T_{Wire} : 경로상의 배선 전체 지연

광범위하게 표현하자면 식(4)로 표현된다. T_{DP}는 는 데이터 경로의 지연이고, T_{CU} 는 컨트롤 유닛의 지연이다.

$$clk = T_{DP} + T_{CU} \quad \text{.....식(4)}$$
$$T_{DP} = T_{TR} + T_{FU}$$
$$T_{CU} = T_{PSR} + T_{DEC} + T_{CL} + T_{NS} + T_{sSR}$$

클럭후보 선택에서는 클럭주기 집합을 결정함에 있어 모든 단계마다 PSDelay를 만족하는 스케줄을 생산한다. 이 과정은 최소한의 자원의 수를 요구하는 클럭 주기를 선택하는 것이 중요하다. 그림3는 n개 상태의 클럭 주기 선택 알고리듬을 나타낸다.

```
Procedure : MinClkPeriod
  Input : a data flow graph DFG, the number of
          states N;
  Output : the minimum clock period;
begin Procedure
  Cstep = 1;
  ComputePathLength(DFG);
  MaxPathLength = delay of the longest path in
   DFG;
  MinClk = MaxPathLength/n
  InsertReadyOps(DFG, Plist);
  while (Plist 0) do
    if Cstep = N then
       schedule all the non-scheduled operations;
       MinClk = maximum state delay;
       Plist = 0;
    else
       op = First(Plist);
       if op is a signal-cycled operator then
        determine chaining or non-chaining;
          schedule op and update MinClk;
          else
          determine the number of cycles of op;
          schedule op and update MinClk;
    endif;
         InsertReadyOps(DFG, Plist);
         Cstep = Cstep + 1;
  endif;
end while;
return MinClk;
end Procedure
```

그림 3. N 개의 상태에서의 최적 클럭 주기 선택 알고리듬

2.3 자원 예측

자원 예측은 후보 클럭에 의해 요구되는 자원의 양과 타입의 수를 평가할 수 있는 과정이다. 이 과정은 모든 단계에서 연산들을 동시에 고려해야하고 또한 파이프 단계를 통한 자원 고유를 고려해야한다. 그림 4에서와 같이 각 DFG1과 DFG2의 두 파이프 단계로 되어있다. 주어진 클럭 주기와 상태의 수를 가지고 , 먼저 각 연산의 시간 프레임을 계산한다. ASAP와 ALAP 스케줄링의 차에 의한 시간 프레임을 구한다.

그림4는 자원 예측을 예시를 나타낸다.

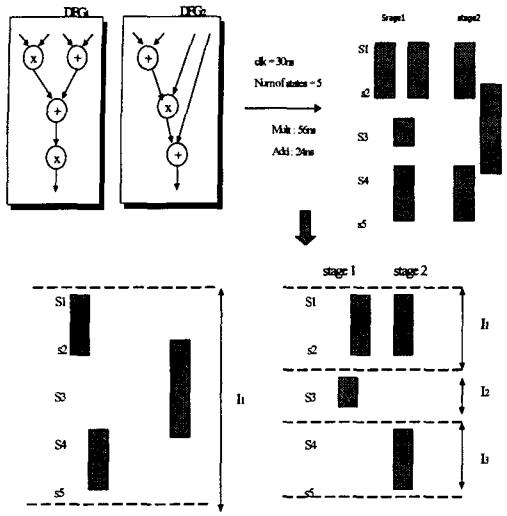

그림 4. 자원 예측을 가정한 예시

3. 실 험 결 과

실험 결과에서는 SUN SPARC 워크스테이션 상에서 C 언어로 클럭 주기 선택 알고리듬을 수행하였다. 본 알고리듬의 효율성을 입증하기 위해서 시간 제약 스케쥴링인 Force-directed 스케쥴링을 선택하여 비교하였다. 또한 다양한 벤치마크를 통해 제안한 알고리듬의 우위성을 입증하였다. 표1에서와 같이 3개의 상위 레벨 벤치마크 중 AR 필터와 HAL 필터는 동일한 결과를 얻을 수 있었고, EL 필터인 경우 2파이프 단계에서는 10.5% , 3 파이프 단계에서는 무려 33.4 % 클럭 주기 감소를 얻을 수 있었다.

표 1. 클럭 주기 선택 알고리듬과 FDS의 비교

Exam.	# of stage	PSDelay (ns)	FDS		본 논문	
			clk(ns)	자원	clk(ns)	자원
AR	2	150	16.6	4A,5M	16.6	4A,5M
	3	100	16.6	6A,8M	16.6	6A,8M
EF	2	300	37.2	3A,2M	33.3	4A,2M
	3	200	33.3	5A,2M	22.2	5A,3M
	4	150	16.6	5A,2M	16.6	5A,2M
HAL	2	150	50	1A,1S 2M	50	1A,1S 2M

4. 결 론

본 논문은 저전력 RTL 설계를 위한 최적 클럭 주기 선택 알고리듬을 제안한다.

제안된 알고리듬은 소비 전력을 최소화하기 위해 시스템의 성능을 증가시킨 다음 전압을 낮추어 처리량을 유지시키는 방법으로 본 논문에서는 파이프라인 기법을 이용해서 성능을 최대한으로 한 뒤 소비 전력을 줄이는 방법을 선택한다. 또한 자원 공유를 최대로 하기 위해 최적 클럭 주기 선택 알고리듬을 적용하여 효율성을 입증하였다. 실험 결과에서 EL 필터인 경우 최대 33.4% 클럭 주기 감소를 얻었다.

향후 연구과제로는 다중 클럭 주기를 고려한 알고리듬을 적용하는 것이 수행되어야 하겠다.

참 고 문 헌

[1] S. Devadas, S. Malik, " A Survey of Optimization Techniques Targeting Low Power VLSI Circuit," in Proc. 32nd DAC, pp. 242-247, June 1995.

[2] A. Chandrakasan, T. Sheng, and R. Brodersen, " Low Power CMOS Digital Design," Journal of Solid State Circuits, vol. 27, no. 4 pp. 473-484, April 1992.

[3] R. Hartley, "Behavioral to Structural Translation in a Bit-Serial Silicon Compiler," IEEE Trans. CAD, vol. 7. no. 8, Aug. 1988, pp.877-886

[4] A. Chandrakasan, R. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits," IEEE Proceedings, vol. 83, no. 4, April 1996, pp.498-523

[5] A. Ghosh, " Estimation of Average Switching Activity in Combination and Sequential Circuits", in Proc. 29th DAC, June 1992, pp.253-259

[6] C. Knapp, P.J. Kindlmana, and M.C. Papaefthymiou. Implementing and Evaluating Adiabatic Arithmetic Unit. In Proceeding of the IEEE 1996 Custom Integrated Circuit Conference. May 1996.