

# 최적 모듈 선택 아키텍처 합성을 위한 전력 감소 Force-Directed 스케줄링

최지영\*, 김희석\*\*

## Low Power Force-Directed scheduling for Optimal module selection Architecture Synthesis

Ji-young Choi\*, Hi-seok Kim\*\*

요 약

본 논문은 최적 모듈 선택 아키텍처 합성을 위한 전력 감소 Force-directed 스케줄링을 제안한다.

제안한 전력 감소 스케줄링은 행위 수준 언어를 입력으로 스위칭 활동(switching activity)을 고려하여 기존의 FDS 스케줄링을 저 전력으로 고려한 FDS\_LP 알고리즘을 수행한다. 제안한 FDSL\_LP 알고리즘은 스위칭 활동을 최소화 하는 동적 파워를 포스 개념에 적용하여 전력 감소를 수행한다. 모듈 선택에서는 전력, 면적, 지연의 매개 변수를 고려하여 최적 모듈 선택 RT 라이브러리를 구축한다. 구축한 RT 라이브러리에서 최적 파라미터를 구하기 위해서 브랜치 앤드 바운드 방법을 사용한 최적 모듈 선택 방법을 제안한다.

비교 실험에서는 최적 모듈 선택을 고려한 제안한 FDS\_LP 알고리즘과 기존의 FDS 알고리즘간의 전력 차이를 비교하여 최대 23.9 % 까지 전력 감소를 얻을 수 있다.

키워드 : 모듈선택, FDS, 스케줄링, FDS\_LP, 브랜치 앤드 바운드

ABSTRACT

In this paper, we present a reducing power consumption of a scheduling for module selection under the time constraint.

The proposed low power scheduling executes FDS\_LP considering low power to exist the FDS scheduling by inputted the behavioral language. The proposed FDS\_LP performs lower power consumption with dynamic power which is minimized the switching activity, based on force conception. In the time step of module selection, an optimal RT(Register Transfer) library is composed by exploration of the parameters such as power, area, and delay. To find optimal parameters of RT library, an optimal module selection algorithm using Branch and Bound algorithm is also proposed.

In the comparison and experimental results, The proposed FDS\_LP algorithm reduce maximum power saving up to 23.9% comparing to previous FDS algorithm

**Keyword** . module selection, FDS, Scheduling, FDS\_LP, Branch and bound

### I. 서론

최근 접어들어 다양한 형태의 디지털 시스템의 휴대화에 대한 수요가 급증하게 되었다. 특히, 휴대

용 장치에 있어서 크기나 중량과 더불어 중요한 요소를 차지하는 배터리 용량은 집적 회로가 소모하는 전력에 직접적으로 비례하게 되었다. 그리고 전력 손실이 큰 회로에 대한 packaging/cooling 비용

\* 제천기능대학 정보통신설비과 (cjy03@kopo.or.kr), \*\* 청주대학교 전자공학과 (khs8391@chongju.ac.kr)

논문번호 KICS2004-05-005, 접수일자 2004년 5월 10일

\*\* 본 연구는 과학기술부 · 한국과학재단 지정 청주대학교 정보통신연구센터의 지원에 의한 것입니다

의 상승으로 저 전력 소모를 위한 디지털 시스템 설계의 필요성이 요구되고 있다. 그러나 최근까지는 이러한 추세와는 달리 디지털 시스템의 설계에 있어서 주된 관심은 대상 회로의 속도의 증가와 성능 측면에 중점을 두어 왔으며, 상위 수준에서의 저 전력 소모를 지원하기 위한 연구가 최근 활발 진행되고 있는 실정이다. 일반적으로 CMOS 회로에서의 전형적인 전력 소모의 요인으로 스위칭 활동 (switching activity), 누설전류(leakage current), 폐 회로 전류(short-circuit current) 등에 의하며, 이들 중 스위칭 동작에 의한 전력 소모가 약 90% 이상으로 가장 큰 비중을 차지한다[1-4] 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. 앞서서도 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. [5-8] 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.

광범위하게 보면 저 전력 시스템의 전력은 모든 설계 프로세스에서 최적화 될 수 있다.[9] (시스템, 알고리즘, 구조, 로직, 회로, 프로세싱 기술 등) 본 논문에서 표현된 동작적 레벨 전력 최적화에 주요 점을 둔다. 알고리즘 레벨의 변환은 전력의 최적화에 사용되어 왔다.[10] 또한 기능 연산자의 활동의 최소화를 위한 절차는 많이 제안되어 왔다.[11-13] 또한 다른 행위적 레지스터 수를 줄이는 방법, 레지스터의 스위칭 활동을 줄이는 방법, 효율적인 레지스터 할당을 통해 전력 소모를 줄인다.[12] 그러므로 그중 전력 소모를 가장 효율적인 방법은 공급

전압을 최소화를 통한 방법이라고 할 수 있다. 또한 이에 따른 회로 지연의 파이프라인 및 병렬처리에 의해 보상되어진다.[2] 처리 능력을 유지하는 다른 방법은 가변 공급 전압에서 조작한 자원을 사용하는 것입니다.[15-16] 시간 제약 스케줄링은 ILP (Integer Linear Programming) 와 FDS (Force Directed Scheduling) 가 기본적인 스케줄링 방법이다. 이중 ILP 를 이용한 방법은 전체적인 최소 전력 소모를 얻을 수 있으나 시간 복잡도가 크므로 소규모의 DFG (Data Flow Graph)의 대상으로 사용되었다. FDS 는 ILP 와는 달리 휴리스틱 방법으로 시간 제약 조건하에서 사용하는 기능 연산자 수를 최소화 하는 것이 목적이므로 저전력을 고려한 스케줄링도 FDS 방법이 기존의 ILP 방법보다 효율적이다. 본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 최적 모듈 선택 아키텍처를 위한 전력감소 FDS\_LP 에 대해 기술하고, 3장에서는 최적의 모듈 선택 방법을 제시하고 4장에서는 기존의 FDS 와 본 논문에서 제안한 FDS\_LP 를 비교 실험하였고, 마지막으로 결론으로 구성되어 있다

## II. 최적 모듈 선택 아키텍처 합성을 위한 전력 감소 Force-Directed 스케줄링

최적 모듈 선택을 위한 아키텍처 합성을 위한 전력 감소 스케줄링은 첫 번째 단계에서는 전력 감소를 위한 스케줄링으로 스위칭 활동 (switching activity)을 고려하여 스케줄링 및 할당 바인딩을 수행한다. 이 단계에서의 스케줄링의 주안점은 기존의 FDS 스케줄링을 저전력으로 고려한 FDS 방법을 적용한다. 두 번째 단계에서는 전력, 면적, 지연의 매개 변수의 상호 상관관계를 고려하여 RT 라이브러리를 구축한다. RT 라이브러리는 전력, 면적, 지연의 매개 변수를 고려하여 구축한다. 제안한 최적 모듈 선택 방법은 브랜치 앤드 바운드 접근 방식을 이용하여 최적화 방법을 채택하였다. 모듈 선택 방법은 전력, 면적, 지연을 동시에 고려하여 설계 조건에 최적화하였다 또한 전력 부분에서는 전체적인 전력 소모를 스위칭 동작과 다중 전압의 전력 소모도 고려하였다. 일반적으로 모듈 선택과 저전력 라이브러리는 별도로 취급하여 구축되어 왔으나 본 논문에서는 이런 모든 조건을 고려하여 최적의 조건을 만족할 수 있는 RT 라이브러리를 탐색하는 방법을 나타내었다. 기존의 전력 라이브러리는 전력 예측 모델을 기반으로 스위칭 동작양이 가장 적은

것을 선택하여 구축하였다. 그림 1은 RT 라이브러리를 통한 상위 레벨 합성 과정을 나타낸다.

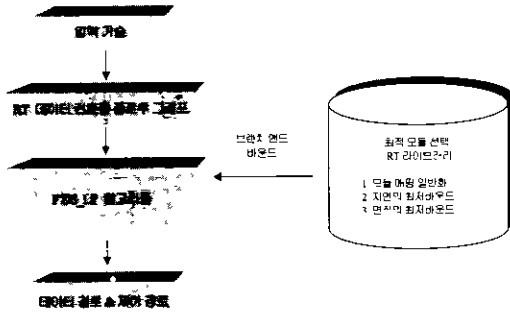


그림 1 최적 모듈 선택에 의한 상위 레벨 합성

### 2.1 기존의 FDS 알고리즘

FDS 알고리즘은 ILP 기반의 스케줄링 알고리즘은 달리 휴리스틱 방법으로 시간 제약 조건하에서 사용하는 기능 연산자 수를 최소화 하는 것을 목적으로 한다. 먼저 주어진 DFG에 대해 ASAP 와 ALAP를 수행하여 각 연산에 대해 Si 와 Li 를 얻은 후에, 각각의 연산에 대한 타임 프레임과 연산 종류에 따른 분포 그래프를 구성한다. 타임 프레임으로부터는 각각의 연산들이 할당될 수 있는 제어 단계의 범위를 알 수 있는데 이것을 이동도라고 한다. 그리고 이 분포 그래프를 바탕으로 제어 단계 j 에서 각각의 연산의 self-force를 다음과 같이 정의한다.

$$Self-Force(j) = \sum_{i \in S_j} Force(i) \quad \dots \text{식(1)}$$

$$Force(i) = DG(i) \cdot x(i) \quad \dots \text{식(2)}$$

$$DG(i) = \sum_{Op \in OpType} Prob(Op, i) \quad \dots \text{식(3)}$$

식(2)에서 DG(i) 는 제어 단계 i에서 해당 연산 종류가 존재할 확률이고, 식(3)의 Prob(Op, i) 는 연산 종류가 OpType 인 연산이 제어 단계 i 에 존재할 확률이다. 그리고 x(i)는 연산을 제어 단계 i 에 두었을 때 해당 DG(i)의 변화율로서 i 와 j가 같으면 양의 값이고 다르면 음의 값이다. 하나의 연산

을 특정 제어 단계에 할당하면 CDFG 상에 연결된 다른 연산들의 타임 프레임은 변화시키므로, CDFG 상에 연결된 다른 연산의 영향을 고려하여 식(1)의 자체포스 외에 선행포스와 후속포스를 함께 구하여 전체포스를 정의한다. 선행포스는 현재 진행되고 있는 연산의 스케줄링에 영향을 받는 상위 제어 단계의 연산들에 대한 포스이고, 후속포스는 반대로 하위 제어 단계의 연산들에 대한 포스이다.

$$전체포스(i) = 자체포스(i) + 선행포스(i) + 후속포스(i)$$

....식(4)

이와 같이 각각의 연산에 대해 total-force를 구한 후, 전체포스가 가장 작은 제어 단계에 연산을 할당한다. 하나의 연산이 제어 단계에 할당되면 새로운 타임 프레임과 분포 그래프가 생성되고 이를 토대로 다시 각각의 연산에 대해 전체포스를 구한다. 이와 같은 과정을 마지막 연산이 스케줄링될때 까지 반복한다. FDS 는 ILP 기반 스케줄링과는 달리 휴리스틱 방법이므로 그 수행 속도가 빠르고 각 연산 종류마다 제어 단계상에서 존재하는 확률이 일정하도록 스케줄링하기 때문에 스케줄링 결과도 상당히 우수하다.

### 2.2 새로운 FDS\_LP 알고리즘

이 과정에서는 일반적인 스케줄링 방식인 FDS를 저 전력 차원에서 FDS\_LP 알고리즘을 제안하였다. 제안한 FDS\_LP 알고리즘은 자원 공유 확률의 측정에 의해서 스위칭 동작을 최소화한다. 기존의 FDS에서의 포스를 스위칭 동작모델로 변환하기 위해 포스 방정식  $F = k \cdot x$  으로부터 첫 번째 모듈 내부의 스위칭 동작을 스프링 장력 상수 K에 반영한다. 기존의 FDS는 개별적 자원(type-distribution)을 위한 연산 사이의 경쟁과 연관이 있다. 또한 적합한 자원 공유 선택의 확률은 변위(displacement) x 와 상관있다. 기존의 FDS 방법에서는 균형적인 스케줄링을 측정하기 위해 이동도의 변화율이 적은 타임스텝안의 연산자를 스케줄링하기 위해서 연산자 간의 이동도 변화를 고려하였다. 그러나 본 연구에서는 부분적 스케줄링 정보를 이용하기 위해 기존의 FDS와 달리, 연산자 사이의 적합한 자원 공유 확률을 결정 알고리즘 초기에 임계 연산자로 스케

줄링한다. 표 1은 포스 매개 변수의 분석의 차이점을 설명한다. 그리고 그림 2는 제안한 FDS\_LP 알고리즘을 나타낸다.

표 1 포스 매개 변수 분석

알고리즘	스프링상수	변위	포스
FDS	개별적인 연산자 타입	연산자 동도의 변화	연산자의 이행행성
FDS_LP	모듈의 위칭이 동속	소함 선택의 이화률	동적 위칭

```

Algorithm FDS_LP(DFG G, primary input trace VT)
Profile DFG with VT
latency_constraint ← critical_pathlength(G)
determine_time_frame(ops ∈ G)
if for all op ∈ G which are on the critical_path
    op.tstep ← critical_timestep
while unscheduled ops exist do
    if(op.tstep = UNKNOWN) then
        for each t ∈ time_frame(op) do
            Evaluate_vaild_combination(op,t)
            for each c ∈ vaild_combination do
                p ← probability_product(c)
                swact ← switching_activity(c)
                force(op,t,c) ← p * swact /* f=kx 를 이용 */
                max_force ← determine_max_force(op,t,c)
            for each op ∈ G do
                for each t ∈ time_frame(op) do
                    force_norm(op,t,c) ← normalize
                    force(op,t,c) write max_force
                    compute mean(op,t), std_dev(op,t),
                    skew(op,t) of force_norm(op,t)
                    total_force(op,t) ← mean(op,t) + std(op,t)
                    + skew(op,t)
                    min_total_force(op,t) ←
                    determine_minimum_total_force(op,t)
                op.tstep ← t and update G
            endwhile
        return G
    end algorithm
    
```

그림 2 FDS\_LP 알고리즘

그리고 반복적인 자원이 계층적으로 구성되어 있을 때의 공유된 자원에서의 스위칭 활동(switching activity)를 미치는 영향을 예를 설명한다. 그림 3은 공유된 자원에서의 스위칭 활동과 자원공유 영향을 나타낸다.

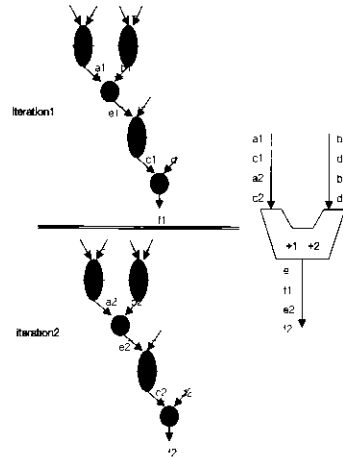


그림 3 공유된 자원에서의 스위칭 활동과 자원공유 영향

### III. 최적의 모듈 선택 방법

모듈을 선택하는 과정은 물리적 정전 용량을 고려하여 적절한 하드웨어를 할당하는 상위 수준 합성 단계이다. 기능연산자, 메모리, 레지스터, 멀티플렉서 그리고 버퍼들은 모듈 선택 단계에서 라이브러리로부터 선택된다. 일반적으로 상대적 물리적 정전 용량이 큰 유닛들은 동작 속도가 매우 중요시되는 회로에서 고려 대상이다. 저 전력 설계의 아키텍처 합성은 전력, 면적, 지연의 매개 변수에 상호 독립적이기 때문에 최적화 문제가 복잡하다. 최적 모듈 선택 방법으로 브랜치 앤드 바운드 접근 방식을 제안하였으며, 이는 동시에 전력, 면적 지연을 동시에 고려하여 최적화를 수행한다. 일반적인 모듈 선택 및 저전력 설계는 별도로 취급하여 기존의 전력 라이브러리는 전력 예측 모델을 기반으로 스위칭 동작량이 가장 적은 쪽을 선택하여 구축하였으나 본 연구에서는 모든 조건을 고려하여 최적의 조건을 만족할 수 있는 RT(Register Transfer)라이브러리를 구축하였다.

#### 3.1 모델 및 표현 (Model and Representation)

##### 1) RT 라이브러리

제안된 알고리즘을 측정하기 위해 라이브러리를

표2와 같이 표시된 것을 채택하였다. 각 아키텍처에 적합한 연산의 전압, 지연, 면적, 전력을 합한 정보를 표현한다. 평균 스위치 활동은 PRESTO 를 이용하여 일반화 하였고 컴포넌트는 0.6 마이크로 3.3 V VTI표준 셀 CMOS 라이브러리를 이용하였다. 또한 이와 같은 컴포넌트들은 단지 2 입력 NAND 와 NOR 그리고 인버터만으로 합성되었다. 그림 4 에서와 같이 같은 가산 연산이라 하더라도 각 구조에 따라 모듈 선택을 달리 선택하여야 한다. 그림 4 는 RT 컴포넌트 선택 아키텍처를 나타낸다.

표 2. RT 라이브러리

	Module Type	Voltage (V)	Delay (ns)	area ( $\mu^2$ )	power (mW)
-	ripple carry adder(RCA)	3.3	150	1258	5.4
		5	80	1258	22.7
	carry look ahead adder(CLA)	3.3	80	6598	10.5
		5	40	6598	37.3
*	booth multiplier (Booth)	3.3	320	16455	30.7
		5	145	16455	84.0
	array multiplier (Array)	3.3	160	32090	143.1
		5	100	32090	295.6

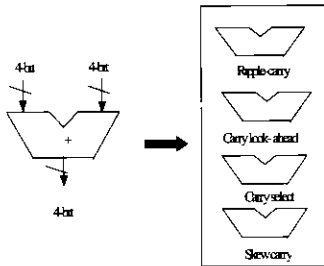
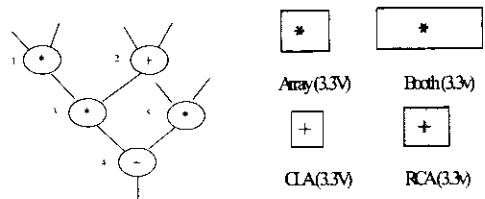


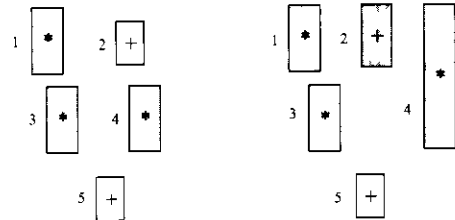
그림 4 RT 컴포넌트 선택 아키텍처

또한 아키텍처 선택 과정에서 예를 들어 그림 5 와 같은 (a) DFG 와 (b) 모듈 타입별 아키텍처의 주어진 제약으로 전력 및 특성이 있을 경우 가산 (adder) 연산을 리플 캐리 가산(ripple carry adder), 캐리 룩 어 헤드 가산(carry-look ahead adder), 스쿼 캐리 가산(skew carry adder) 등 여러 아키텍처 구조의 가산 모델링을 라이브러리로 선택하고 재사용 하는데 있어 앞서서도 언급했지만 예를 들어, 보통의 공급 전압 하에서의 RCA 는 조금 속도가 느리기는 하지만, 적은 물리적 정전 용량으로 인해 CSA

에 비해 에너지 측면에서 효율적이다. 하지만 저 전압 환경으로 감에 따라 그것은 속도 제한을 도저히 만족할 수 없게 된다. 이러한 경우는 물리적 정전 용량이 다소 크더라도 CSA를 사용하는 목적이 여기에 있다. 이렇듯 주어진 전력 감소 제약 특성에 맞게 나타낸 아키텍처 선택을 (c) 와 (d)에서 비교할 수 있다. 일반적으로 스케줄링 제어시스템(Control Step)은 동일하고 전력 및 면적 면에서 (d)의 아키텍처가 저 전력에 유리한 구조를 나타낸다. 그림 5 는 모듈에서의 아키텍처 선택을 다음과 같이 나타낸다.



(a) DFG (b) 모듈 타입별 아키텍처



delay 400, power 450.3 area 70778 delay 400, power 332.8 area 56401

(c) 아키텍처 1 (d) 아키텍처 2

그림 5 아키텍처 선택

2) 모듈 선택 용어 정의

(정의 1) 데이터 플로우 그래프 (DFG) : 방향성 그래프  $G(V, E)$  로서  $V$  는 연산의 집합이다.

(정의 2) 타입 : 연산자  $V$  를 나타낸다.  $t(v)$  는 연산을 수행하는 함수이다

(정의 3) 모듈 . 가산기는 덧셈으로 승산기는 곱셈 등으로 연산의 종류를 하나 또 그 이상으로 수행할 수 있는 연산을 수행할 수 있으며 ALU는 모두 수행한다.

(정의 4) 모듈 라이브러리 : 모듈의 집합으로  $M$ 으로 정의된다

(정의 5) 모듈 지원 집합 연산을  $v$ , 연산종류를  $\tau$  ( $v$ )로 표현하며,  $\Psi(v)$ ,  $\Psi(\tau)$ 은 연산을 수행할 수 있는 모든 모듈의 집합이다.

예를 들어, 위의 표의 모듈 라이브러리는  $\Psi(+)$  = {RCA, CLA},  $\Psi(*)$  = {ARRAY, BOOTH} 이다.

(정의 6) 모듈 매핑:  $f : X \rightarrow M$ ,  $X \subseteq V$ , 은  $X$ 안의 각 연산들은 수행하고 모듈을 할당한다  $X \subset V$  이면  $f$ 는 부분적 모듈 매핑이라 불리우며  $X = V$  이면 완전 모듈 매핑이라 한다.

(정의 7) 주어진 모듈 매핑  $f$ 는 면적, 지연시간, 전력 소모 등에 관한 비용 함수를 가지고 있다.  $Area\_LB(f)$ ,  $Delay\_LB(f)$ ,  $Power\_LB(f)$  은 상대적으로  $f$ 의 면적, 지연, 전력비용의 하한 경계를 나타낸다.

그러므로 모듈 선택은 주어진 최적의 목적을 가진 모듈 라이브러리로 주어진 DFG을 위한 최적의 모듈 매핑을 찾는 문제이다.

### 3) 모듈 선택 알고리즘 방법

전절에서 기술하였듯이 지연과 면적을 고려한 전체적인 최적 전력 감소 즉 면적, 지연, 전력을 최적화한 파라미터를 고려한 아키텍처를 설계하기 위해서는 모듈 선택의 전체 설계 영역을 조사해야 한다. 앞에서 정의한 본 연구에서는 모듈 선택의 조건에 따라 브랜치 앤드 바운드 알고리즘을 사용하여 최적의 모듈 선택을 휴리스틱 방법으로 모듈을 선택해서 최적의 결과를 구하고자 한다. 그러나 모듈 선택 영역에서 DFG 전체에 대하여 모듈을 검색하는 것은 바람직하지 못하다. 그러므로 최적 모듈 선택을 위한 효율적인 제거 기법이 중요하다. 모듈 매핑 수행 후, 지연상의 최저 바운드(Lower Bound)와 면적 및 전력 소모의 최저 바운드를 수행한다. 그림 6은 제안된 모듈 선택 및 합성 알고리즘을 나타낸다

```

Main()
{
    Bound ← A vector of Maximum numbers
    v ← An operation which has no predecessor
    X ← {v}, f ← ∅
    Call ModuleSelection (X, f, v, Bound)
    For each f in the feasible configuration list
    {
        If LB(f) - Bound > δ Delete f /* 다시제거 */
    }
    For each f in the feasible configuration list

```

```

{
    Perform Scheduling and Allocation on f
    Find the optimal architecture
}
target architecture ← the optimal architecture
}

```

```

Proc ModuleSelection(X, f, v, Bound)
{
    If X = ∅ Stop
    Repeat
    {
        If all the value of  $\Psi(v)$  then
        {
            X ← X | {v}; Return /* 되돌아옴 */
        }
        Else {
            g(v) ← next un-tried value of  $\Psi(v)$ 
            f ← f ⊗ g(v) /* 모듈 매핑 */
            If LB(f) - Bound > δ then Continue
            Else {
                X ← X ∪ {v} {
                    If f is a full mapping
                    {
                        feasible configuration list ← f.
                        If LB(f) < Bound then Bound ← LB(f)
                    }
                    Else {
                        v ← next un-visited operation in V
                        Call ModuleSelection(X, f, v, Bound)
                    }
                }
            }
        }
    }
    Until( all the value of  $\Psi(v)$  have been tried)
}

```

그림 6. 제안된 모듈 선택 및 합성 알고리즘

#### ① 모듈 매핑 일반화

제안된 모듈 매핑은 깊이 우선순위 기법으로 바운드를 향상시킬 수 있는 매핑으로 최적 설계에 적용한다. 그림 7은 주어진 표2의 모듈 라이브러리를 가지고 모듈 선택 영역을 결정하는 방법을 나타낸 것으로 연산자의 동작 전압을 3.3 V로 가정하였다. 그림에서 보는 바와 같이 완전 매핑 노드 A가 구해지고 노드 A의 비용은 바운드를 가진다. 노드 B는 노드 A와 같은 지연을 가진 매핑이나 면적과 전력비용이 A보다 작다. 그러므로 모듈 매핑에서 바운드는 B로 갱신된다 노드 C와 노드 D는 좀 더 큰 지연을 가짐으로서 삭제되고 노드 E는 면적 비용이 크기 때문에 삭제된다.  $\delta$ 는 검색 에지의 면적, 전력비용과 바운드 값과의 오차 한계를 설정하여 설계 목적에 더 적합한 노드를 구하고자 설정한 값이다. 예를 들어, F는 면적이 다소 크더라도 좀 더 나은 전력비용을 가지고 있다. 그림 7은 탐색된 모듈 선택 영역을 나타낸다.

노드 A . Delay<sub>LB</sub> = 400, Area<sub>LB</sub> = 70778, Power<sub>LB</sub> = 450.3  
 노드 B Delay<sub>LB</sub> = 400, Area<sub>LB</sub> = 55143, Power<sub>LB</sub> = 337.9

노드 C : Delay<sub>LB</sub> = 470  
 노드 D : Delay<sub>LB</sub> = 480  
 노드 E : Delay<sub>LB</sub> = 400, Area<sub>LB</sub> = 72036, Power<sub>LB</sub> = 445.2  
 노드 F : Delay<sub>LB</sub> = 400, Area<sub>LB</sub> = 56401, Power<sub>LB</sub> = 332.8

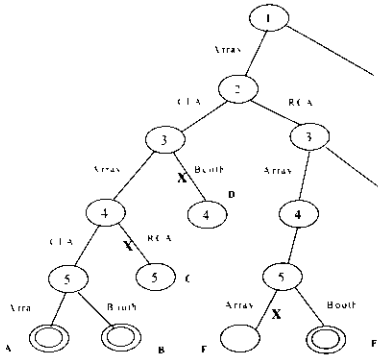


그림 7. 탐색된 모듈 선택 영역

② 지연의 최저바운드

연산자들을 전부 방문했을 때 매핑 함수 f의 지연의 최저 바운드는 깊이 우선 탐색(Depth First Search) 방법에 의해 모든 연산이 가장 빠른 시간으로서 지연시간으로 정의하고 finish\_time(v)는 휴리스틱 알고리즘이다. 그림 8은 지연의 최저바운드 알고리즘을 나타낸다.

```

Pred(v) = { u | u ∈ f-1, (u, v) ∈ E }
delay_lb ← 0
while ∃ v ∈ f-1 that finish_time(v) has not defined
{
    If Pred(v) = ∅ finish_time(v) ← delay(τ(v))
    Else { finish_time(v) ← delay(τ(v)) + Max {
        finish_time(u) | u ∈ Pred(v) }
    }
    delay_lb ← Max(delay_lb, finish_time(v))
}
    
```

그림 8 지연의 최저바운드

③ 면적과 전력 소모의 Lower Bound

면적과 전력소모의 최저바운드를 측정하기 위해 주어진 매핑에 위해 요구되어진 최소의 자원의 수를 평가하는 것이 중요하다. R<sub>w</sub> 는 f에 요구되어진 타입 w의 최소의 모듈 수를 정의한다. 면적과 전력 소모에 대한 최저바운드 비용함수는 식(4)와 식(5)

로 주어진다.

$$Area\_LB(f) = \sum_{w \in f} R_w * area(w) \dots \text{식(4)}$$

$$Power\_LB(f) = \sum_{w \in f} \sum_{all that f(u)=w} R_w * power(w) \dots \text{식(5)}$$

또한 R<sub>w</sub> 측정은 최소 간격 중첩 알고리즘(minimal interval overlap algorithm)을 사용한다. 그리고 ASAP(v)와 ALAP(v)를 정의하고 시간간격 [a,b]에서 연산의 부분집합 X는 최소 간격 중첩으로 정의된다. 그림 9은 자원의 최저 바운드 알고리즘이다.

```

For each module type w which ∈ f
{
    S ← { ASAP(v) | f(v) = w }
    L ← { ALAP(v) | f(v) = w }
    Sort S and L in ascending order R_w ← 0
    For each interval [ s1 , l1 ]
    ( s1 ∈ S, l1 ∈ L, s1 < l1 )
    { R_w ← Max( R_w | O[X, [s1 , l1]] / delay(τ) )
    }
}
    
```

그림 9 자원의 최저바운드

IV. 실험 결과

제한한 최적 모듈 선택 아키텍처 합성을 위한 전력 감소 알고리즘은 C 언어로 SUN SPARC 워크스테이션에서 구현되었다. 본 논문에서 비교 실험한 벤치마크는 첫 번째 Testexample 으로 테스트 스케줄링 알고리즘을 위한 전형 모형 DFG로 5개의 덧셈과 2개의 뺄셈, 15개 에지로 구성되어 있는 예제이고, 두 번째는 FIR(Second Order Finite Impulse Response filter)필터로 5개의 곱셈과 4개의 덧셈, 19개 에지로 구성된다. 세 번째는 IIR(Second Order Infinite Impulse Response filter)필터로 5개의 곱셈과 4개의 덧셈, 19에지로 구성되어 있다. 마지막으로 Paulin는 6개의 곱셈과 3개의 덧셈, 15개 2개의 뺄셈, 25에지로 구성되어 있다. 본 알고리즘은 DFG로 입력을 받아 기존의 FDS 방식과 본 논문에서 제안한 FDS\_LP 알고리즘을 비교 평가하였다.

표3는 시간 제약과 사용된 자원의 수를 나타내는 표이다. 여기에서는 결과는 기존의 FDS 나 제안한 FDS\_LP 알고리즘은 동일한 제어 스텝의 결과를 얻었다 기능 연산자(functional unit)와 멀티플렉서(multiplexer)간의 절충(tradeoff)을 제외하고는 레지스터(register) 개수는 동일한 결과 값을 보였다.

표 3 시간제약에 따른 제어 스텝과 사용된 자원의 수 비교

벤치마크	제어스텝		사용된 자원					
	FDS	FDS_LP	FDS			FDS_RPC		
			Reg	FUs	Mux	Reg	FUs	Mux
Testexample	4	4	8	3	7	8	4	5
FIR	5	5	10	3	6	10	4	8
IIR	5	5	10	5	10	10	6	9
Paulin	4	4	14	5	11	14	6	10

또한 데이터 환경은 ARMA (Autoregressive moving average) 신호를 사용해서 모델링되었다. 이 모델링은 상위 레벨 파워 추정 및 최적화에 기초로 제공된다. ARMA 모델은 일반화된 데이터 스트림안의 특정한 정확성을 설명하는데 사용되어지며 널리 응용되어지고 있다[28][29]. 제안된 알고리즘의 비교실험에서는 2개의 ARMA 방정식 SIG 1 :  $Y(n) - 0.5x(n-1)$  과 SIG 2 :  $Y(n) + 0.1x(n-1)$ 을 데이터 환경으로 채택하였다. ARMA 방정식은 오디오와 스피치 신호에 사용된다.[28] 표 4과 표 5는 SIG1 과 SIG2 2개의 ARMA 방정식에 따른 스위칭 동작을 나타낸다. 동일한 환경 안에서의 실험에도 불구하고 벡터 집합의 길이가 다양한 결과 값을 나타낸다. 비교에서도 나타나는 바와 같이 FDS 와 FDS\_LP 스위칭 동작이 표 4에서는 최대 24.8% 전력감소를 보였고, 표 5에서는 최대 18.0% 전력 감소량을 나타냈다.

알고리즘 벤치마크	FDS	FDS_LP	Reduction(%)
Testexample	131.3	98.7	24.8%
FIR	160.8	145.9	9.2%
IIR	229.3	208.4	9.1%
Paulin	386.2	324.6	15.9%

표 4 SIG 1 환경으로부터의 FDS 와 FDS\_LP 전력 소모  
표 5 SIG 2 환경으로부터의 FDS 와 FDS\_LP 전력 소모

알고리즘 벤치마크	FDS	FDS_LP	Reduction(%)
Testexample	67.5	55.3	18.0%
FIR	80.2	72.8	9.2%
IIR	133.3	123.3	7.5%
Paulin	213.1	204.8	3.8%

표6은 FDS\_LP 기반으로 설계 단계의 최대 전력 감소량은 23.9%를 얻을 수 있었고, 반면에 최소 전력 감소량은 10.9% 얻었다. 또한 모든 벤치마크 평균 전력 감소량은 16.4%가 되었다. 표6은 FDS 와 FDS\_LP 설계 단계별 전력 감소 결과를 나타낸다.

표 6 FDS와 FDS\_LP 설계 단계별 전력감소결과

벤치마크	Design	Reg	FUs	MUX	Control
Testexample	23.9%	5.7%	59.3%	61.4%	15.2%
FIR	11.5%	8.3%	7.4%	26.3%	8.5%
IIR	10.9%	12.2%	8.0%	13.8%	8.1%
Paulin	19.3%	2.0%	39.0%	44.2%	12.3%
평균 세이빙	16.4%	7.1%	28.5%	36.7%	11.1%

### V. 결론

본 논문은 최적 모듈 선택 아키텍처 합성을 위한 전력 감소 Force-directed 스케줄링을 제안한다.

제안된 전력 감소 스케줄링 알고리즘은 기존의 FDS 스케줄링을 전력 감소를 고려한 FDS\_LP 알고리즘을 제안하였다. 제안한 FDS\_LP 스케줄링은 스위칭 동작을 최소로 하는 동적 파워를 포스 F의 개념에 적용하여 전력 감소를 수행하였다. 포스 F는 모듈의 스위칭 동작과 조합 선택의 확률의 곱으로써 전력을 표현하였다. 즉,  $F = K \cdot X$  에서 K 는 스프링 장력을 나타내고 변위 X 는 최저 평균과 표준 편차, 스쿼를 포함한 동적 파워에 대한 가장 최적의 값이다. 이와 같은 변위는 평균 스위칭 활동이 가장 최소인 값에 해당이 된다. 또한 모듈 선택에서는 전력, 면적, 지연의 매개 변수를 고려하여 최적 모듈 선택 RT 라이브러리를 구축하였다. 최적 모듈



선택 방법은 브랜치 앤드 바운드 접근 방식을 이용하여 면적, 지연, 전력의 최적 파라미터로서 모듈 선택을 휴리스틱 방법으로 최적의 결과를 얻은 모듈 선택을 찾았다. 비교 실험에서는 최적 모듈 선택을 고려한 제한한 FDS\_LP 알고리즘과 기존의 FDS 알고리즘의 간의 전력 차이를 비교하여 최대 전력 감소량은 23.9 %를 얻을 수 있었고, 반면에 최소 전력 감소량은 10.9% 얻었다. 모든 벤치마크 평균 전력 감소량은 16.4 %가 되었다.

본 연구의 기대 효과로는 고성능 저전력 데이터 및 제어 경로 설계의 구현을 위하여 DSP, 마이크로컨트롤러, ASIC, Network 설계에 효과적으로 적용할 수 있을 것으로 사료된다.

### 참 고 문 헌

- [1] R. Hartley, "Behavioral to Structural Translation in a Bit-Serial Silicon Compiler," IEEE Trans. CAD, vol. 7. no. 8, Aug. 1988, pp 877-886
- [2] A. Chandrakasan, R. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits," IEEE Proceedings, vol. 83, no. 4, April 1996, pp.498-523
- [3] A. Chandrakasan et al., "Low-Power CMOS Digital Design," J. Solid-State Circuits, vol 27, no.4, April 1992, pp.473-484
- [4] A. Ghosh, " Estimation of Average Switching Activity in Combination and Sequential Circuits", in Proc. 29th DAC, June 1992, pp.253-259
- [5] P. Landman, " Power Estimation of High-Level Synthesis", in Proc. European DAC, Feb. 1993, pp.361-366
- [6] A. Chandarkasan et al., "HYPER-LP: A System fo Power Minimization Using Architecture Transformation," in Proc. ICCAD, Nov. 1992, pp.300-303
- [7] R. Martin, "Power-Profiler Optimizing ASICs Power Consumption at the Behavioral Level," in Proc 32nd DAC, June 1995, pp.42-47
- [8] J. Chang, "Register Allocation and Binding for Low Power , in Proc. 32nd DAC, June 1995, pp.29-35
- [9] A. chandraksan and R. Brodersen, "Minimizing power consumption in digital CMOS circuit" in Proc. IEEE, vol.83, Apr. 1995, pp.498-523.
- [10] A. chandraksan, M. Potkonjak, R. Mehra, J. Rabaey, and Brodersen, "Optimizing power using transformations," IEEE Trans. Computer-Aided design, vol. 14, pp.12-31, Jan.1995.
- [11] A. Dasgupta and R. Karri, "Simultaneous scheduling and binding for low power minimization during microarchitecture synthesis," in Proc. Int. symp. Low-Power design, Apr 1995, pp.69-74.
- [12] A. Raghunathan and N. K. Jha, "Behavioral synthesis for low power," in Proc. IEEE design Automation conf, 1995.
- [13] e. Musoll and J. Cortadella, "High-level synthesis techniques for reducing the activity of functional unit," in Proc. Int. Symp. Low Power Design, 1995, pp. 99-104.
- [14] H.Singh and D. D. Gajski, "A Design Methodology for Behavioral Level Power Exploration Implementation and Experiments", Technical Report 397-28, University of California, Irvine, 1997
- [15] J.-M. Chang and M. Pedram, " Energy minimization using multiple supply voltage", IEE E Trans. VLSI Syst., vol. 5, Dec. 1997.
- [16] M.C. Johnson and K. Roy," Datapath scheduling with multiple supply voltages and level converters", ACM Trans. Design Automat. Electron. Syst., vol. 2, no.3, pp.227-248, July 1997

최 지 영(Ji-young Choi)



e-mail : [cjy03@kopo.or.kr](mailto:cjy03@kopo.or.kr)

74년 4월 12일생

97년 2월 세명대학교

전자계산학과 졸업(이학사)

99년 2월 세명대학교

전산정보학과 졸업(이학석사)

2004년 8월 청주대학교

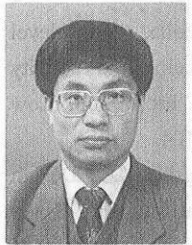
전자공학과 박사졸업

(공학박사)

2002년 3월 ~ 현재 세천기능대학 조교수

<관심분야> CAD, 알고리즘, 상위레벨합성, 저전력

김 희 석(Hi-seok Kim)



e-mail :

[khs8391@chongju.ac.kr](mailto:khs8391@chongju.ac.kr)

54년 12월 23일생

77년 2월 한양대학교

전자공학과 졸업(공학사)

80년 2월 한양대학교

전자공학과 졸업(공학석사)

85년 8월 한양대학교

전자공학과 졸업(공학박사)

87년 9월 ~ 88년 9월 미국

University of Colorado at Boulder 객원 교수

96년 8월 ~ 97년 7월 미국

University of Callifornia at Irvine 객원 교수

81년 3월 ~ 현재 청주대학교 전자공학과 교수

<관심 분야> CAD, 컴퓨터 구조, 컴퓨터 알고리즘