
CPLD 조건식을 위한 RTL 바인딩

김재진* · 윤충모**

A RTL Binding Technique for CPLD constraint

Jae-Jin Kim* · Choong-Mo Yun**

본 논문은 2006년도 서일대학 학술연구비 지원에 의해 연구되었음

요 약

본 논문에서는 CPLD 조건식에 위한 RTL 바인딩을 제안하였다. HDL로 기술된 회로에 대해 스케줄링을 수행한 후 모듈 연산 간격을 고려하여 합당한 모듈을 선택하여 할당을 수행한다. 할당을 수행한 후 주어진 조건식에 맞도록 CPLD를 선정한다. 할당된 결과의 모듈을 CPLD 내부의 CLB의 맞도록 부울식을 분할하고, 최적의 CLB를 사용하여 회로를 구현할 수 있는 바인딩 알고리즘을 제안하였다. 16 비트 FIR 필터로 실험한 결과 알고리즘을 적용하기 전보다 작은 크기의 CPLD로 회로 구현이 가능하였으며, 가산기의 경우 알고리즘을 적용하지 않았을 때 내부 사용율은 8.45%이었으나 알고리즘 적용한 결과 61.88%로 내부 사용율이 증가되는 결과를 나타내었다.

ABSTRACT

In this paper, a RTL binding technique for CPLD constraint is proposed. Allocation processing selected module consider the module calculation after scheduling process for circuit by HDL. Select CPLD for constrain after allocation. A Boolean equation is partitioned for CLB by allocated modules. The proposed binding algorithm is description using optimum CLB within a CPLD. The proposed algorithm is examined by using 16 bit FIR filter. In the case that applicate the algorithm, the experiments results show reduction in used CLB.

키워드

RTL 바인딩, CPLD 조건식, CPLD, CLB

I. 서 론

상위수준 논리합성(High-level synthesis)에서 HDL(Hardware Description Language)로 기술된 회로를 ASIC(Application Specific Integrated Circuit)나 FPGA(Field Programmable Gate Array), CPLD(Complexity Programmable Logic Device)등을 사용하여 널리 회로를

구현하고 있다.

FPGA를 이용하여 회로를 구현하고자 할 경우 실행되는 순서는 주어진 조건식에 따라서 할당(Allocation)과 스케줄링(Scheduling), 바인딩(Binding)을 수행한 후 플레이스(place)와 라우트(route)를 하게 된다. 그러나 이러한 방법은 FPGA 소자의 특성을 고려하지 않고 수행되므로 사용자가 원하는 회로 구현이 되지 않는 경우가 발생되며

* 극동정보대학 컴퓨터정보과 교수

** 서일대학 정보전자전공 교수

이러한 경우에 처음부터 다시 실행을 시켜야 하는 단점이 있다. 따라서 이러한 단점을 보완하기 위해 회로를 구현하고자 하는 FPGA 소자의 조건식을 정보로 가지고 있으면서 레이아웃(layout)을 고려하여 회로를 구현할 수 있는 레이아웃을 고려한(layout-driven) 방법이 제안되었다.[1-4]

그러나 CPLD의 경우, 회로 구현용 툴들은 많이 제공되고 있으나 상위수준 논리합성에서 CPLD의 조건식을 고려하여 회로를 구현하는 방법은 아직 제안되지 않았다.

따라서 본 논문에서는 상위수준 논리합성에서 CPLD의 조건식을 고려하여 회로를 구현할 수 있는 RTL 바인딩 알고리즘을 제안하였다.

그림 1은 본 논문에서 제안한 CPLD 조건식에 위한 RTL 바인딩 알고리즘의 전체 순서도 이다.

II. CPLD 조건식에 위한 RTL 바인딩

CPLD를 이용하여 회로를 구현하는 일반적인 CAD 툴들은 HDL에서 기술된 회로 구성을 스케줄링과 바인딩을 통해 하나의 CPLD에 맵핑(mapping)하였다.[5-8] 그러나 이러한 맵핑은 회로의 지연 시간과 CPLD의 내부 지연시간에 대한 비교가 없었으며, 입출력 핀의 수와 부울식의 조건식을 고려하지 않고 수행하였다.

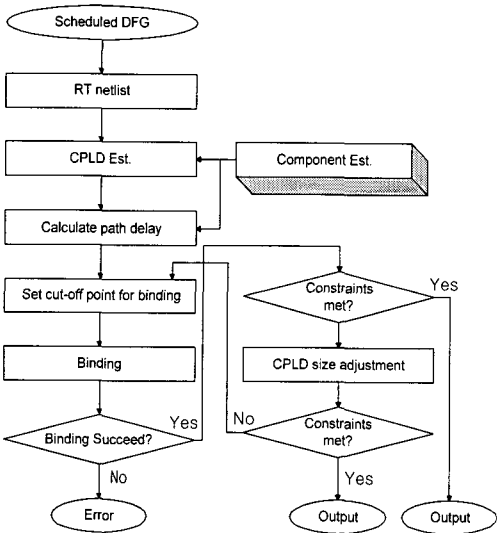


그림 1. CPLD 조건식에 위한 RTL 바인딩의 전체 순서도
Fig. 1. Flow chart of a RTL binding technique for CPLD constraint

```

#define num16 num<16,10>

#define a0 -0.001953125
#define a1 0.003906250
#define a2 -0.007812500
#define a3 0.01953125
#define a4 -0.06640625
#define a5 0.7500000

func main(In : num16) Out : num16 =
begin
    Acc1=num16(In@10*a0);
    Acc2=num16(In@9*a1)+Acc1;
    Acc3=num16(In@8*a2)+Acc2;
    Acc4=num16(In@7*a3)+Acc3;
    Acc5=num16(In@6*a4)+Acc4;
    Acc6=num16(In@5*a5)+Acc5;
    Acc7=num16(In@4*a4)+Acc6;
    Acc8=num16(In@3*a3)+Acc7;
    Acc9=num16(In@2*a2)+Acc8;
    Acc10=num16(In@1*a1)+Acc9;
    Out=num16(In*a0)+Acc10;
end;
    
```

그림 2. 16 비트 FIR 필터의 입력 파일
Fig. 2. 16bits FIR filter input file

본 논문에서 제안한 CPLD 조건식에 위한 RTL 바인딩은 HDL로 기술된 회로의 할당과 스케줄링, 바인딩을 수행한 후 하나의 CPLD로 회로 구현이 가능한가를 측정 후 가능하다면 미리 사용자가 지정한 조건식과 비교하여 일치할 경우 회로를 구현하게 된다.

만약 바인딩의 결과 하나의 CPLD로 회로 구현이 불가능한 경우는 CPLD의 조건식을 고려하여 회로의 할당과 스케줄링, 바인딩을 2개 이상의 CPLD에 구현할 수 있도록 새로 수행함으로써 두 개 이상의 CPLD로 회로를 구현할 수 있도록 하는 기술이다.

본 논문에서 제안한 CPLD 조건식에 위한 RTL 바인딩의 예제로 16 비트 FIR 필터를 선정하였다. 그림 2는 HDL로 기술된 16비트 FIR 필터의 입력 파일을 나타내었다.

그림 3은 그림 2의 16 비트 FIR 필터의 입력 파일을 Hyper에 입력하여 얻어낸 스케줄링 결과이다.

스케줄링 결과 2개의 승산기(Multiplier)와 하나의 가산기(Adder)로 할당되었다.

할당된 결과와 회로의 입출력 개수와 부울식의 조건식을 토대로 CPLD의 조건식과 비교하여 CPLD 내부에 회로가 어떻게 구현되는가에 대한 정보를 갖게 된다.

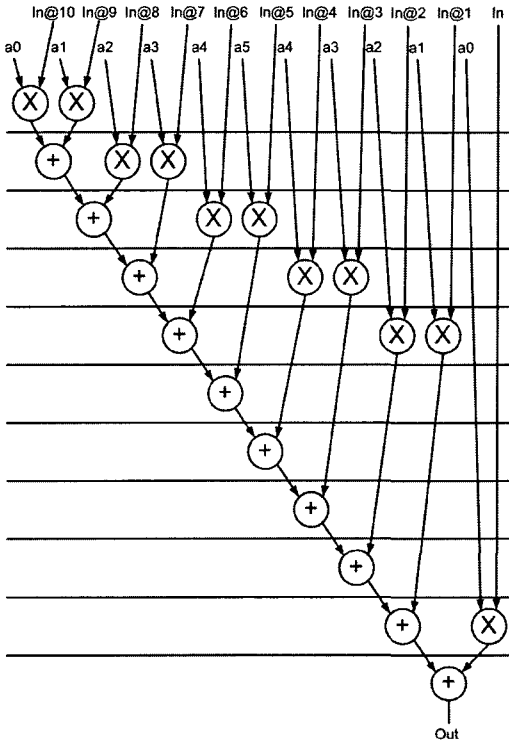


그림 3. 16 비트 FIR 필터의 스케줄링 결과
Fig. 3. Result of 16 bits FIR filter schedule

그림 4는 할당된 결과와 CPLD 내부에서 회로 구현의 예상도이다.

그림 4와 같이 2개 이상의 CPLD에 회로를 분할 하기 위해서는 회로의 동작 특성과 CPLD의 조건식과 주어진 조건식을 고려하여 스케줄링과 바인딩을 수행하여야 한다.

스케줄링은 회로의 전체 지연 시간과 연산자의 지연 시간, 모듈의 지연 시간을 고려하여 가장 시간적인 요소가 일치되는 모듈을 찾게 된다.

전체 지연 시간의 계산은 식 1에 의해 계산된다.

$$t_d = t_{pr} + t_{sr} + t_{im} + t_{fu} + t_{om} + t_{wrm} + t_{wmf} \quad (식 1)$$

- t_{pr} : Propagation delay
- t_{sr} : 레지스터의 setup time
- t_{im} : 입력 MX의 지연시간
- t_{fu} : FU의 지연시간
- t_{om} : 출력 MUX의 지연시간
- t_{wrm} : 레지스터에서 입력 MUX까지의 지연시간
- t_{wmf} : 입력 MX에서 FU까지의 지연시간

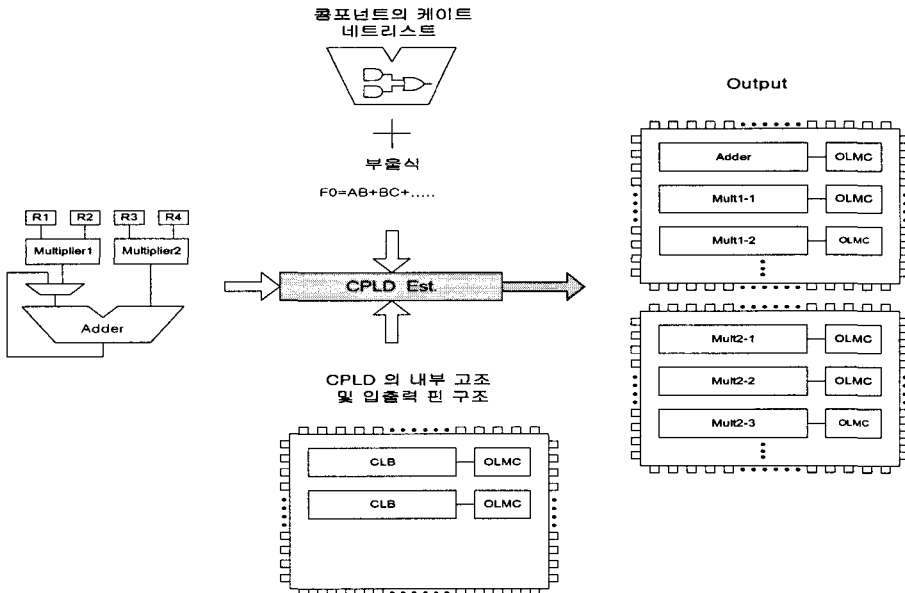


그림 4. 할당된 결과와 CPLD 내부에서 회로 구현의 예상도
Fig. 4. Allocated result and estimation of CPLD circuit process constrain

모듈을 찾는 방법은 그래프 형식의 연산자 수행 간격(OEI : Operation Execution Interval)와 모듈 수행 간격(MEI : Module Execution Interval)를 비교하여 적당한 수행 시간을 갖는 모듈을 찾게 된다.

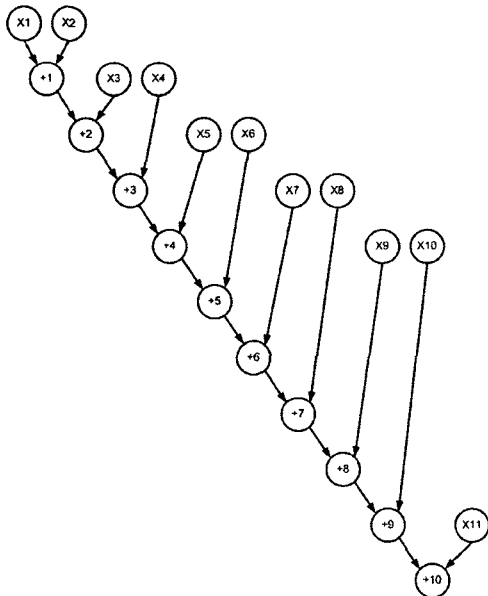
그림 5는 OEI와 MEI를 비교하여 적당한 모듈을 찾는 방법을 나타낸 그림이다. 그림 5에서 굵은 선으로 표현된 MEI의 모듈을 선택하여 사용한다.

식 1을 이용하여 전체의 지연시간을 계산하고 그림 5와 같은 방법으로 사용할 모듈이 결정되면, 주어진 조건식의 시간과 입출력에 맞게끔 분할을 해야 한다.

분할은 CPLD 내부의 CLB는 3개에서 10개 정도의 OR 텀을 가지고 있으므로 하나의 출력을 형성하기 위해 2개 이상의 CLB를 사용하여야 하므로 CPLD의 전체 지연시간이 조건식에서 주어진 시간의 1/2이 되는 CPLD를 선정한다.

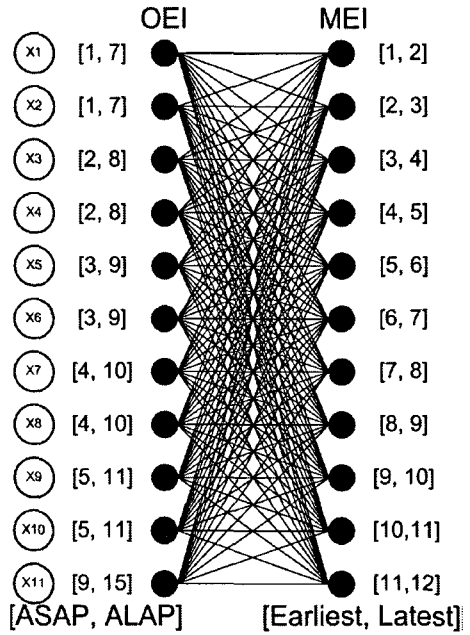
$$t_d = \frac{1}{2} t_{con}$$

t_{con} : constraint에서 지정한 시간



시간적 제한 : 15 사이클(Cycle)
 자원 : 2개의 가산기와 2개의 곱셈기

(a) DFG



(b) 곱셈기 분할

그림 5. 모듈을 찾는 방법
 Fig. 5 Method of finding module

CPLD가 선정되면 선택된 모듈의 부울식을 추출하여 선택된 CPLD의 CLB 크기에 맞도록 부울식을 분할한다.

분할된 부울식을 선택된 소자에 바인딩하는 방법은 그림 6과 같은 알고리즘에 의해 수행된다.

```

Binding(S_fu, Ivar, Ovar, N, M,
Result_allocation)
// S_fu : 선택된 FU
// Ivar : 입력 변수
// Ovar : 출력 변수
// N : S_fu의 수
// M : T_CPLD의 CLB 수
T_CPLD; // 선정된 CPLD
CLB; // 선정된 CPLD의 CLB 수
GBE ← Generate_BE(S_fu);
// 선정된 모듈의 부울식 생성
    
```

```

begin
for(j=1 to M)
    
```

```

if(GBE < CLB_size)
  // CLB_size : CLB의 OR텀수
  VirtualBind(S_fu, CLBi);
else if(GBE > CLB_size)
  PGBE ← PARTITION(GBE, CLB_size);
  // PGBE : CLB의 크기에 맞게 분할하여
  //   생성된 부울식
  VirtualBinding(S_fu, CLBi);
if( i+1 ≤ N)
  success = Binding(S_fu, N, M);
  if(success) then
    return(True);
  else
    UnBinding(S_fu, CLBi);
  end if;
else
  ActualBinding;
  return(True);
end if;
end for;
return(False);
end

```

그림 6. 바인딩 알고리즘
Fig. 6. binging algorithm

III. 실험 결과

본 논문에서 제안한 CPLD 조건식에 위한 RTL 바인딩의 예로 16비트 FIR 필터를 선정하였다.

스케줄링과 할당의 결과는 앞에서 기술하였다

16 비트 FIR 필터는 2개의 곱셈기와 2개의 가산기로 구성되어 있다.

곱셈기와 가산기를 CPLD로 구현하기 위해 ALTERA사에서 제공되는 CPLD를 사용하였으며 조건식은 제한 시간을 100ns로 지정하였고, 입출력 핀의 수는 250 개로 선정하였다.

그 결과 하나의 곱셈기를 구현하기 위한 소자로 FLEX10K30이 선정되었으며, 하나의 가산기를 구현하기

위한 소자로 EPX8160이 선정되었다.

16 비트 곱셈기는 입력 32개 출력 33개로 입출력 핀은 65개를 사용하는데 32번째 연산 결과 출력인 CAL31의 경우 OR 텀의 수가 91개로 출력 핀에 연결된 CLB 외에 89개의 CLB를 더 사용하여야 하며 총 1308개의 CLB를 더 사용하여야 한다.

FLEX10K30은 1728개의 CLB로 구성되어 있어 곱셈기 구현에 적합하다.

또한 가산기의 경우 입출력 핀은 108핀을 사용하며 32개의 다른 CLB를 사용하여야 한다.

EPX8160은 160개의 CLB로 구성되어 있어 가산기를 구현할 수 있다.

실험 결과는 표 1에 나타내었다.

표 1. 실험 결과
Table. 1 Experiment result

예제명	조건식	모듈명	알고리즘 미적용		알고리즘 적용	
			선정된 CPLD (총CLB수)	사용된 CLB수	선정된 CPLD	사용된 CLB수
16 비트 FIR 필터	150 ns 250 핀	곱셈기	FLEX10K40 (2304)	1832	FLEX10K30 (1728)	1341
		가산기	FLEX10K10 (576)	146	EPX8160 (160)	99

IV. 결 론

본 논문은 상위수준 논리합성에서 CPLD 조건식에 위한 RTL 바인딩 알고리즘을 제안하였다.

제안한 알고리즘은 HDL로 기술된 회로에 대하여 스케줄링을 한 후 모듈 연산 간격을 고려한 모듈을 선택하고, 스케줄링과 할당을 수행한 후 주어진 조건식에 맞는 CPLD를 선정한다.

또한 할당된 모듈을 CPLD 내부의 CLB의 조건식을 고려하여 부울식을 분할하고 최적의 CLB를 사용하여 회로를 구현하였다.

제안한 알고리즘에 16 비트 FIR 필터를 예제로 선정하여 ALTERA사에서 제공되는 CPLD를 이용하여 회로를 구현한 결과 알고리즘을 적용하기 전보다 작은 CPLD로 회로 구현이 가능하였으며, 회로 구현에 필요한 CLB의 수가 적었고, 가산기의 경우 알고리즘을 적용하지 않았을

때 내부 사용율은 8.45%이었으나 알고리즘 적용한 결과 61.88%로 내부 사용율이 증가되었다.

참고문헌

- [1] C. Ramachandran, F. J. Kurdahi, D. Gajski, V. Chaiyakul, A. Wu, "Accurate Layout Area and Delay Modeling for System Level Design" Proc. ICCAD'92, NOV. 1992
- [2] M. Xu, F. J. Kurdahi "Chip Level Area and Timing Estimation for Lookup Table Based FPGAs" Tech. Report #95-31, UCI, Aug. 1995
- [3] N. Dutt, C. Ramachandran, "Benchmarks for the 1992 High Level Synthesis Workshop", UCI, 1992
- [4] Ashutosh Mujumdar, Minjoong Rim, Rajiv Jain, Renato De Leone, "BITNET : An Algorithm for solving The Binding Problem" 7th International Conference on VLSI Design, pp. 163-168, 1994
- [5] 윤충모, 김희석, "시간적 조건에서 실행 시간을 개선한 CPLD 기술 매핑 알고리즘 개발", 한국 OA 학회 논문집 vol 4권 3호, pp. 35-46, 1999
- [6] Jae-Jin Kim, Hi-Seok Kim, Chi-Ho Lin, "A New Technology Mapping for CPLD under the time constraint" ASP-DAC, pp.235-238, January 2001.
- [7] 김재진, 이관형, "상관관계에 위한 CLB 구조의 CPLD 저전력 기술 매핑 알고리즘", 한국컴퓨터정보학회 논문집 제10권 제2호, pp.49-57, 2005
- [8] 김재진, 이관형, "시간제약 조건과 면적을 고려한 효율적인 CPLD 기술 매핑", 한국컴퓨터정보학회 논문집 제10권 제3호, pp. 11-18, 2005

저자소개



김재진 (Jae-Jin Kim)

2003년 2월 청주대학교 전자공학과
공학박사

2001년~현재 극동정보대학 컴퓨터정
보과 조교수

※관심분야: VLSI 및 CAD 설계



윤충모 (Choong-Mo Youn)

2000년 2월 청주대학교 전자공학과
공학박사

현, 서울대학 정보전자전공 교수

※관심분야: VLSI 및 CAD 설계