

# 저전력 SoC을 위한 동적 주파수 제어 시스템의 FPGA 프로토타입 설계

## (FPGA Prototype Design of Dynamic Frequency Scaling System for Low Power SoC)

정은구<sup>†</sup>  
(Eun-Gu Jung)

다이아나 마르쿨레수<sup>\*\*</sup>  
(Diana Marculescu)

이정근<sup>\*\*\*</sup>  
(Jeong-Gun Lee)

**요약** 하드웨어 기반 동적 전압 및 주파수 제어 시스템은 전역 비동기 지역 동기 시스템 설계 방식을 이용하여 동종의 멀티 코어 혹은 이종의 멀티 코어 시스템을 저전력으로 설계하기 위한 핵심 기술 중의 하나이다. 본 논문에서 하드웨어 기반 동적 주파수 제어 시스템의 FPGA 프로토타입 설계를 위해서 동적 주파수 제어기를 제안하고, 이를 FIFO 기반 멀티코어를 이용한 소프트웨어 정의 무선 설계와 네트워크 온 칩 기반의 하드웨어 MPEG2 인코더 설계에 적용하였다. 기존의 단일 주파수 시스템에 비해서 소프트웨어 정의 무선 설계의 경우 성능이 5.9% 하락하였지만, 전력소모는 78% 감소하였다. MPEG2 인코더 설계의 경우에 성능은 0.36% 하락하고 전력소모는 29.1% 감소하였다.

**키워드** : 동적 전압 및 주파수 제어, 프로그래밍 디바이스, 시스템 온 칩, 저전력, 전역 비동기 지역 동기

**Abstract** Hardware based dynamic voltage and frequency scaling is a promising technique to reduce power consumption in a globally asynchronous locally synchronous system such as a homogeneous or heterogeneous multi-core system. In this paper, FPGA prototype design of hardware based dynamic frequency scaling is proposed. The proposed techniques are applied to a FIFO based multi-core system for a software defined radio and Network-on-Chip based hardware MPEG2 encoder. Compared with a references system using a single global clock, the first prototype design reduces the power consumption by 78%, but decreases the performance by 5.9%. The second prototype design shows that power consumption decreases by 29.1% while performance decreases by 0.36%.

**Key words** : dynamic voltage and frequency scaling, FPGA, System-on-Chip, low power, globally asynchronous locally synchronous

## 1. 서론

하드웨어 기반 동적 전압 및 주파수 제어(hardware based dynamic voltage and frequency scaling)는 전역 비동기 지역 동기 시스템(globally asynchronous locally synchronous system)의 저전력 설계를 위한 유망한 기술 중 하나이다[1,2]. 이 기술은 각 로컬 모듈이 실시간으로 모듈 자신이 필요로 하는 성능에 맞추어 전압 및 주파수를 하드웨어로 제어함으로써, 성능의 손실을 최소화 유지시키면서 전체 시스템의 전력소모를 최소화할 수 있게 해 준다. 이러한 기법을 FPGA 개발 보드를 이용하여 설계하기 위해서는 전압 및 주파수의 실시간 제어를 위한 하드웨어 지원이 필요하다. 현재, 전압 제어는 FPGA 개발보드에서 지원하지 않고 있으며, DLL/PLL을 이용한 주파수 제어도 최근에 소개된 고성능 FPGA에서만 제한적으로 지원되고 있어 대부분의 FPGA 개

· 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2006-352-D00146)

† 비 회 원 : ETRI 부설 연구소 연구원  
egjung@ensec.re.kr

\*\* 비 회 원 : Carnegie Mellon University 전기 컴퓨터 공학과 교수  
dianam@cmu.edu

\*\*\* 정 회 원 : 한림대학교 컴퓨터공학과 교수  
jeonggun.lee@hallim.ac.kr

논문접수 : 2009년 7월 17일

심사완료 : 2009년 9월 2일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨터의 실제 및 레터 제15권 제11호(2009.11)

발 보드에서는 클럭 배분기를 이용하여 제한된 주파수 범위 내에서 동적 주파수 제어가 가능하다[3,4].

본 논문에서 생산자-소비자 모델[1]에 기반을 둔 하드웨어 DFS의 FPGA 프로타입 설계를 제안하고 이를 FIFO기반 멀티 코어 시스템을 이용한 소프트웨어 정의 무선(Software Defined Radio) 응용 프로그램과 네트워크 온 칩(Network-on-Chip) 기반의 하드웨어 MPEG2 인코더 시스템에 적용하였고, 성능 및 전력소모에 대해서 단일 클럭 시스템과 비교하였다. 이는 [1,2]에서 언급된 동적 주파수 제어 아키텍처의 확장된 일이다.

2장에서는 하드웨어 DFS 아키텍처에 대해서 설명하고, 3장에서는 제한된 FPGA를 위한 하드웨어 DFS 아키텍처의 설계에 대해서 논의한다. 4장에서는 2가지 시스템에서 적용된 하드웨어 DFS에 대해서 설명하고 5장에서 결론을 맺는다.

## 2. Hardware DFS Architecture

생산자-소비자 모델[1]에 기반을 둔 하드웨어 DFS 아키텍처는 그림 1과 같이 모듈, 클럭 제어기, FIFO로 구성되어 있다. 각 모듈 블록은 FIFO 버퍼를 통해서 데이터를 전송하고 full/empty 신호와 모듈의 write/read data ready 신호에 따라서 full/empty stall 신호를 발생시킨다. 예를 들어서 모듈 A의 생산자는 full 상태의 FIFO 버퍼에 새로운 데이터를 전송하기 원하는 순간에 full stall 신호를 발생시킨다. 그 이외의 경우에는 생산자는 full stall 신호를 발생시키지 않는다. 이러한 원리는 FIFO 버퍼의 empty 신호와 모듈의 read data ready 신호에 기반을 두어 empty stall 신호를 발생시키는데도 적용된다. 이때, 처리량 제약(throughput-constrained) 시스템인 경우에는 실선의 full/empty stall신호를 사용하고, 지연시간 제약(latency-constrained) 시스템인 경우에는 점선의 full/empty stall 신호를 사용한다.

클럭 제어기 블록은 식 (1)[1]에 의해서 새로운 주파수를 결정하여 모듈 A에 공급한다. 식 (1)은 모듈 A의 full stall 신호와 모듈 B의 empty stall 신호, 그리고 현재의 주파수와 샘플 기간을 이용하여 아래의 식 (1)을

통해서 결정된다.

$$f_{new} = \begin{cases} f_{cur} \times S & \text{if } S_f > S_e \\ f_{cur} + S & \text{if } S_e > S_f \end{cases} \quad S = 1 - \frac{|S_e - S_f|}{T_{sample}}$$

$f_{new}$  : new frequency  
 $f_{cur}$  : current frequency  
 $S$  : scaling factor  
 $S_e$  : number of clock cycles for empty stall signal  
 $S_f$  : number of clock cycles for full stall signal  
 $T_{sample}$  : sample period

## 3. FPGA를 위한 Hardware DFS Architecture

### 3.1 FPGA용 DFS 아키텍처

하드웨어 DFS 아키텍처를 FPGA 개발 보드를 이용하여 구현할 때, 그림 2와 같이 32비트 마이크로프로세서인 MicroBlaze[5]와 stall\_gen 블록이 모듈 블록을 대신하고, Fast Simplex Link(FSL) 인터페이스와 FSL 버퍼로 구성된 FSL가 FIFO 버퍼를 대신하게 된다. MicroBlaze 마이크로프로세서는 full/empty stall 신호를 발생하기 위한 여분의 포트를 가지고 있지 않기 때문에, 추가적인 FSL 인터페이스를 이용하여 write data ready와 read data ready 신호를 발생시키고, 이와 함께 FIFO 버퍼의 full/empty신호를 이용하여 stall\_gen 블록이 full/empty stall 신호를 발생시킨다. 클럭 제어기 블록과 stall\_gen 블록은 직접 구현하였고, 나머지 블록은 Xilinx IP 라이브러리를 이용하여 구현하였다.

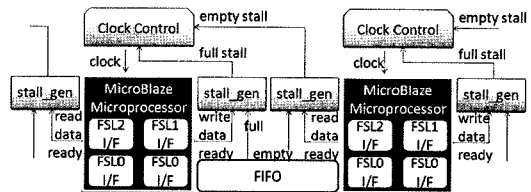


그림 2 FPGA용 DFS 아키텍처(처리량 제약 시스템인 경우)

### 3.2 클럭 제어기 아키텍처

클럭 제어기 블록은 인접한 모듈의 처리시간과 제어하고 있는 모듈의 처리 시간이 같아지도록 주파수를 제어한다. 이것은 성능의 손실을 최소한으로 줄이면서 동시에 전력 소모를 줄이는 방법이다.

클럭 제어기 블록은 그림 3과 같이 클럭 제어기 알고리즘을 저장하는 ROM과 8비트 마이크로프로세서인 PicoBlaze[6], 다양한 주파수를 만드는 클럭 배분기(clock divider), 다음 주파수를 선택하기 위한 테이블 값이 들어 있는 주파수 테이블 ROM(search\_freq\_table

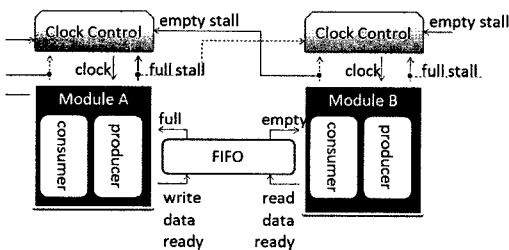


그림 1 하드웨어 DFS 아키텍처

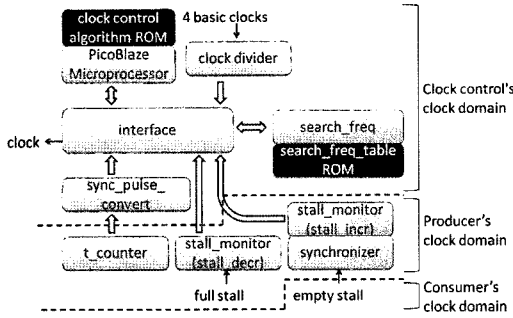


그림 3 클럭 제어기 내부 구조

$$T_{sample} \times \frac{f_{new}}{f_{cur}} = T_{sample} - |S_e - S_f| \quad \text{if } S_f > S_e$$

$$T_{sample} \times \frac{f_{cur}}{f_{new}} = T_{sample} - |S_e - S_f| \quad \text{if } S_e > S_f$$
(2)

표 2 주파수 행렬 테이블

Sample period:		current frequency				
1		20	17.5	...	0.625	0.547
Sample rate	20	1	0.870	...	0.031	0.027
	17.5	0.870	1	...	0.036	0.031
	...	...	...	...	...	...
	0.625	0.031	0.036	...	1	0.870
	0.547	0.027	0.031	...	0.870	1

Upper triangle  
Lower triangle

Sample period:		current frequency				
1024		20	17.5	...	0.625	0.547
Sample rate	20	1024	890	...	32	28
	17.5	890	1024	...	37	32
	...	...	...	...	...	...
	0.625	32	37	...	1024	890
	0.547	28	32	...	890	1024

ROM), stall 신호를 모니터링 하는 stall\_monitor, 클럭 주파수가 다른 영역 간의 동기화를 위한 synchronizer, 샘플 구간을 카운트하는 t\_counter로 구성되어 있다. 여기서 PicoBlaze 마이크로프로세서 기반 구조는 향후 다양한 전압 및 주파수 제어 알고리즘을 비교하기 위해서 사용하였다.

각 stall\_monitor 블록은 샘플 구간동안 발생한 stall 신호를 카운트한다. empty stall 신호는 소비자의 클럭 영역에서 발생되기 때문에, 동기화 블록을 통해서 생산자 클럭 영역으로 동기화 되어야한다. t\_counter 블록은 카운트 값이 샘플 기간과 같을 때 출력 신호를 발생시킨다. 이때, 발생한 신호는 클럭 제어기의 클럭 영역으로 동기화되기 위해서 syn\_pulse\_convert 블록을 사용한다. 표 1과 같이 클럭 분배기는 4개의 기본 주파수로부터 22개의 클럭 주파수를 생성한다.

표 1 22개의 클럭 주파수

Basic input clocks (MHz)	Output clocks (MHz)						
	20	17.5	15	12.5	10	7.5	5
20	20	17.5	15	12.5	10	7.5	5
17.5	17.5	15	12.5	10	7.5	5	3.75
15	15	12.5	10	7.5	5	3.75	2.188
12.5	12.5	10	7.5	5	3.75	2.188	1.094
							0.938
							0.781

stall\_monitor 블록의 결과를 가지고 PicoBlaze 마이크로프로세서는 식 (2)와 같이 새로운 주파수와 현재 주파수의 비율을 계산한다. 이것은 식 (1)의 scaling 요소와 관련되어 있으며, 클럭 제어기 알고리즘을 저장하고 있는 ROM에 어셈블러 코드로 구현되어 있다. 이때, search\_freq 블록은 표 2의 주파수 행렬로부터 얻은 search\_freq\_table ROM값과 주파수 비율 값을 이용해서 새로운 주파수를 결정하게 된다. 예를 들어서 생산자의 처리 시간이 소비자보다 길 경우, 생산자의 처리 시간을 줄이기 위해서 행렬의 위쪽 삼각형 부분이 사용된다. 그렇지 않으면 아래쪽 부분이 사용된다.

클럭 제어기 블록은 FPGA 개발 보드[3]에 사용되는 Xilinx Virtex-II Pro XC2VP30 디바이스를 이용하여 구현하였다. 클럭 제어기 설계에는 474개의 4-input LUT를 사용하였고, 122MHz의 동작 주파수를 기록하였다.

#### 4. FPGA를 위한 Hardware DVFS Architecture

구현된 FPGA용 하드웨어 DFS 아키텍처를 FIFO 버퍼에 기반을 둔 멀티코어 시스템을 이용한 소프트웨어 정의 무선 설계와 Network-on-Chip(NoC)에 기반을 둔 하드웨어 MPEG2 인코더 설계에 적용하였다.

##### 4.1 Software Defined Radio

Software Defined Radio(SDR)[1]는 그림 4와 같이 5개의 블록인 신호를 발생시키는 소스(source), 로우패스필터(low pass filter), 변조파를 복조하는 복조기(demodulator), 로우패스필터를 가진 이퀄라이저(equalizer), 그리고 값을 모으는 싱크(sink)로 구성되어 있다. 각 블록들은 단일 마이크로프로세서에서 동작하는 독립형 응용 프로그램으로 다시 작성되었고, 그것은 클럭 제어를 가진 모듈에 해당된다. 성능을 측정하기 위해서 모니터 모듈이 추가하였고, 실시간으로 각 모듈의 클럭 주파수를 모니터링하기 위해서 Xilinx ChipScope 블록도 추가하였다.

클럭 제어기에 공급되는 4가지 기본 클럭을 생성하기 위해서 그림 5와 같이 클럭 발생기인 DCM 2개와 클럭 분배기인 clock\_div2을 이용하여 20, 17.5, 15, 12.5MHz의 기본 클럭을 생성하였다. 또한, 모니터 모듈에 고정

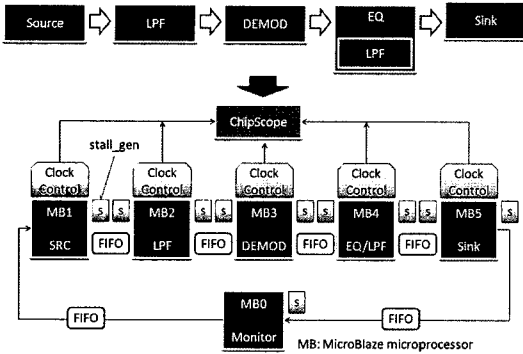


그림 4 DFS을 가진 FIFO기반 멀티 코어 시스템을 이용한 소프트웨어 정의 무선(Software Defined Radio) 설계 구조

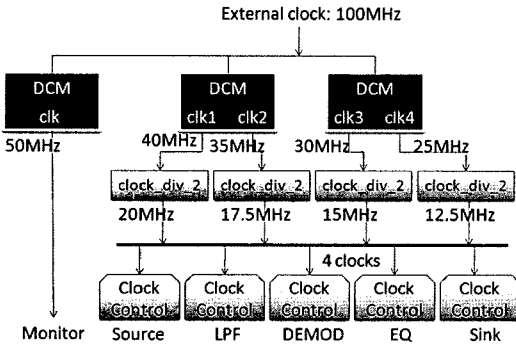


그림 5 DCM을 이용한 4가지 기본 주파수를 생성

된 클럭을 공급하기 위해 DCM을 하나 더 추가하였고 처리량 제한 시스템을 위해서 싱크 블록에는 1.56MHz가 공급되도록 했다.

SDR용 DFS 기능을 지원하는 멀티코어 시스템을 Xilinx EDK와 ISE 캐드툴을 이용하여 설계하였다. 이때, 사용된 개발보드는 Virtex-II Pro XC2VP30 디바이스를 가진 Xilinx XUP Virtex-II Pro 개발보드이다. 설계된 시스템은 14,040 4-input LUT와 123 Block RAMs을 사용하였다.

표 3과 같이 하드웨어 DFS의 다양한 성능을 비교하기 위해서 2가지 시스템을 더 구성하였으며, 이들에 대한 실험값을 표 3에서 보여주고 있다. 첫 번째 시스템인 static\_1은 모든 모듈에 20MHz의 같은 주파수를 공급하는 것이고, 두 번째 시스템(static\_2)은 정적분석[1]에 의해서 구해진 주파수를 각 모듈에 공급하는 것이다. 표 4와 같이 제안된 시스템이 static\_1 시스템에 비해서 5.9%의 성능하락이 있지만 78%의 전력소모를 감소시켰다. 비록 static\_2 시스템에 비해서는 성능과 전력소모에서 약간의 손해를 보았지만, 제안된 DFS 설계를 통해

표 3 3가지 시스템의 클럭 주파수

	Monitor	SOURCE	LPF	DEMOD	EQ	SINK
Static_1(MHz)	50	20	20	20	20	20
Static_2 [1](MHz)	50	20	3.13	1.56	20	1.56
Ours(MHz)	50	20(Max) ~ 0.54(Min)				1.56

표 4 성능 및 전력 소모 측정 결과

	Static_1	Static_2	Ours
Throughput	1	1.017	1.059
Power consumption	1	0.208	0.22

서 많은 노력을 요하는 정적분석을 수행하지 않아도 유사한 성능 및 전력소모를 나타냄을 알 수 있다.

### 4.2 하드웨어 MPEG2 인코더

하드웨어 MPEG2 인코더는 그림 6과 같이 영상 프레임의 종류에 따라서 I 프레임은 코사인 변환 및 양자화(DCT/Q), 허프만 코딩(VLE)으로 부호화하고 P/B 프레임은 움직임 벡터(MV)을 이용한 움직임 보상(MC) 과정을 거쳐 후, DCT 및 VLE를 적용한다. 각 모듈은 Verilog 코드를 이용하여 직접 구현되었고 두 개의 라우터를 이용하여 7개의 모듈을 연결하였다. 이때, DFS를 적용할 모듈은 크게 3가지로 나누었고, 처리량 제한 시스템을 위해서 VLE 모듈의 클럭은 고정된 주파수를 가진다.

NoC에 기반한 하드웨어 MPEG2 인코더의 설계결과 19,201개의 4-input LUT와 111 Block RAM을 사용하였다. 설계된 시스템의 성능 및 전력소모를 측정하기 위

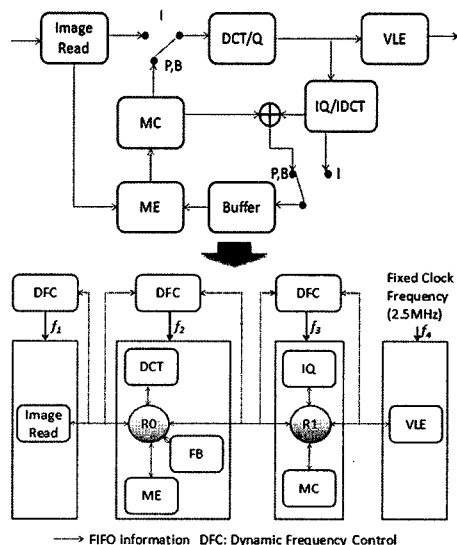


그림 6 DFS 기능을 가진 Network-on-Chip 기반의 MPEG2 인코더 설계 구조

해서 5MHz의 동일한 주파수를 공급하는 레퍼런스 시스템을 만들었다. 동일 주파수를 공급하는 시스템 비해서 DFS를 가진 시스템의 성능은 0.36% 감소하였으나 전력소모는 29.1% 감소하는 이득을 보았다.

## 5. 결론

본 논문에서는 저전력 SoC을 위한 하드웨어 기반 주파수 제어 시스템의 FPGA 프로토타입 설계를 위해서 하드웨어 동적 주파수 제어 시스템을 설계하였고, 이를 FIFO기반의 멀티코어 시스템에 적용하였다. 이때 사용된 응용 소프트웨어는 Software Defined Radio이며, 단일 주파수를 사용하는 시스템에서 비해서 성능에서는 약 5.9% 손실이 있었으나, 전력소모에서 약 78% 이득이 있었다. 또한, 네트워크 온 칩 기반의 하드웨어 MPEG2 인코더에 적용하여 단일 주파수 시스템에서 비해서 성능 손실은 0.36%이지만, 전력소모가 29.1% 감소를 알 수 있었다. 향후 연구 과제로서, 실시간 제약에 따른 DVFS 적용 알고리즘에 대한 연구 개발 및 이를 통한 성능-전력소모 간의 tradeoff 관계에 대한 연구를 진행하고자 한다.

## 참 고 문 헌

- [1] Puru Choudhary and Diana Marculescu, "Hardware Based Frequency/Voltage Control of Voltage Frequency Island Systems," In *Proc. CODES+ISSS*, pp.34-39, 2006.
- [2] Umit Y. Ogras, Radu Marculescu, Diana Marculescu, Eun Gu Jung, "Design and Management of Voltage-Frequency Island Partitioned Networks-on-Chip," *IEEE Trans. on Very Large Scale Integration Systems*, vol.17, no.3, pp.330-341, March, 2009.
- [3] Xilinx, "Virtex-II Pro Development System," UG 069(v1.1) Apr 9, 2008.
- [4] Xilinx, "ML505/ML506/ML507 Evaluation Platform," UG 347(v3.1) Nov. 10, 2008.
- [5] Xilinx, "MicroBlaze Processor Reference Guide," UG081(v9.0) Jan 17, 2008.
- [6] Xilinx, "PicoBlaze 8-bit Embedded Microcontroller User Guide," UG129(v1.1.2) Jun 24, 2008.

### 정 은 구

2000년 경북대학교 전자컴퓨터공학부(학사). 2006년 광주과학기술원 정보통신공학과(석사, 박사). 2006년 광주과학기술원 정보통신공학과 연구전담교수. 2007년 미국 카네기 멜론 대학 박사후 연구원. 현재, 한국전자통신연구원 부설연구소 연구원. 관심분야는 저전력 비동기 회로 설

계, 저전력 VLSI 설계, FPGA 기반 고속 암호 알고리즘 설계



Diana Marculescu

1991년 "Politehnica" University of Bucharest 전자계산학과 석사. 1998년 남가주대학교 컴퓨터 공학과 박사. 현재, 카네기 멜론 대학교 전기컴퓨터공학과 교수. 관심분야는 energy aware computing; 저전력 시스템을 위한 CAD tools, 등



이 정 군

1996년 한림대학교 전자계산학과(학사) 2005년 광주과학기술원 정보통신공학과(석사, 박사). 2005년 캠브리지대학교, 컴퓨터랩, 박사후 연구원. 현재, 한림대학교 컴퓨터 공학과 조교수. 관심분야는 저전력 비동기 회로 설계, 저전력 VLSI 설계, FPGA 기반 설계, 멀티코어 시스템

계, FPGA 기반 설계, 멀티코어 시스템