論文2003-40SC-6-5

단열회로를 이용한 16-bit 저전력 마이크로프로세서의 설계

(A Design of 16-bit Adiabatic Low-Power Microprocessor)

申 暎 濬 **, 李 炳 勳 **, 李 燦 豪 *, 文 龍 *

(Young-Joon Shin, Byung-Hoon Lee, Chan-Ho Lee, and Yong Moon)

요 약

단열회로를 이용한 16-bit 저전력 마이크로프로세서를 설계하였다. 본 논문에서 설계한 마이크로프로세서는 콘트롤 블록, 멀티포트 레지스터 파일, 프로그램 카운터 그리고 ALU로 구성되어 있다. 또한 저전력 단열 프로세서에 필요한 효율적인 4-phase 전원클럭 발생기도 설계하였다. 단열회로는 ECRL(Efficient Charge Recovery Logic)을 기반으로 설계되었고 0.35㎛ CMOS 공정을 이용하여 구현하였다. 단열프로세서와 일반적인 프로세서와 에너지를 비교하기 위해서 CMOS를 기반으로한 프로세서를 설계하여 에너지 비교를 수행하였다. 시뮬레이션 결과 기존의 CMOS 프로세서보다 2.9~3.1배의 에너지 감소효과를 보였다.

Abstract

A 16-bit adiabatic low-power Microprocessor is designed. The processor consists of control block, multi-port register file, program counter, and ALU. An efficient four-phase clock generator is also designed to provide power clocks for adiabatic processor. Adiabatic circuits based on efficient charge recovery logic(ECRL), are designed 0.35㎛ CMOS technology. Conventional CMOS processor is also designed to compare the energy consumption of microprocessors. Simulation results show that the power consumption of the adiabatic microprocessor is reduced by a factor of 2.9~3.1 compared to that of conventional CMOS microprocessor.

Keyword : 마이크로프로세서, 저전력, ECRL, 단열회로

Ⅰ. 서 론

CMOS 기술의 발달로 인한 칩의 고집적화, 프로세서의 고성능화로 인한 열의 발생문제는 현재의 큰 문제점으로 남아있다. 고온의 동작환경에서의 회로의 오동작, 패키지 및 쿨링에 관련된 문제는 해결하기 어려운 문제점으로 남아있는 실정이다. 이러한 문제점들은 현대사회에서 휴대용 컴퓨터, PDA, 이동통신기기의 발달로 인하여 문제점이 더욱 커진다고 볼 수 있다. 동작속도보다는 동작시간이 더욱 중요해지는 상기와 같은 기기에서 배터리 수명연장은 반드시 필요하다. 위와 같은 문제점을 해결하고 더 나은 방법을 제시할 수 있는 단열회로에 대한 연구가 현재 많이 진행되고 있다[1-8]. 단

* 正會員, 崇實大學校 情報通信電子工學部
(School of Electronic Engineering, Soongsil University)
** 學生會員, 崇實大學校 電子工學科
(Dept. of Electronics Engineering Soongsil University)
接受日字:2003年8月10日, 수정완료일:2003年10月20日

열회로는 에너지 복원을 위한 AC형태의 전원을 사용함으로써 에너지 소비를 줄일 수 있는 한가지 방법이다. 본 논문에서는 기본적인 회로부터 단열회로의 방식으로 설계함으로써 기존의 프로세서보다 적은 에너지 소모를 보일 수 있는 단열 마이크로프로세서를 설계하였다. 또한, MIPS 명령어를 기반으로 명령어 세트를 간소화하고 CMOS 프로세서와 Adiabatic 프로세서를 설계하였다. CMOS 마이크로프로세서는 HDL을 이용하여 설계하고, 단열 프로세서의 경우는 full-custom 방식으로 기본적인 회로부터 전체 프로세서에 이르기까지 저전력 회로설계를 기반으로 매크로 블록들을 설계하고 설계된 매크로 블록들을 통합하여 설계하였다. 프로세서로써의 기능을 갖추기 위해 Control Unit, Arithmetic and Logic Unit, Shifter, Program Counter, Register File의 매크로 블록들을 설계하였고[12], 각각의 매크로 블록들을 통합하여 저전력 마이크로프로세서를 구현하였다. 저전력 마이크로프로세서의 경우 AC형태의 전원공급이 필요하기 때문에 실용성을 갖기 위해서는 AC형태의 전원클럭 발생기의 설계도 필요하다. 본 논문에서는 효율적인 전원클럭 발생기도 설계를 하였다[8].

여러 가지의 단열회로들 중에서 ECRL(Efficient Charge Recovery Logic)[7]을 기본으로 하여 설계를 하였다. 레지스터 파일의 메모리셀 이외에는 대부분 ECRL 형태의 회로로 설계하여 적은 에너지 소모를 얻어낼 수 있도록 하였다. 최종적으로 에너지 소모의 비교를 위하여 단열 프로세서를 CMOS 형태의 프로세서로 변경 및 설계하여 두 프로세서간의 에너지를 비교하여 보았다. Ⅱ장에서는 구현한 프로세서를 위한 명령어 세트를 소개하고, Ⅲ장에서는 단열회로의 기본적인 동작과 설계한 저전력 마이크로프로세서의 전체적인 구조, 프로세서를 구성하는 각 매크로블록들의 구조와 동작을 설명하였다. Ⅳ장에서는 단열 마이크로프로세서에 전원클럭을 공급하는 효율적인 전원클럭 발생기의 구조와 동작을 설명하였다. Ⅴ장에서는 실험결과로써 단열회로의 에너지 복원과정을 나타내었으며, CMOS 마이크로프로세서의 명령어 simulation 및 저전력 프로세서의 동작을 나타내었다. Ⅵ장에서는 프로세서를 칩으로 제작한 레이아웃을 나타내었으며, 마지막 Ⅶ장에서는 결론과 현재의 문제점 그리고 향후 개선해야 될 사항에 대하여 정리하였다.

Ⅱ. RISC 명령어 세트 정의

RISC 프로세서를 설계하는데 있어서 고려해야 할 특징은 명령어 세트의 간소화에 있다. Adiabatic 마이크로프로세서와 CMOS 마이크로프로세서를 간소화된 명령어 세트로 설계를 하였다. 단열회로를 이용한 설계가 CMOS회로 보다 설계가 복잡한 것을 고려하여 구조를 간소화하기 위해 기존의 일반 16-bit RISC 프로세서보다 더 간소화된 명령어 세트를 사용하였다. <그림 1>에서 보면 기존의 RISC 프로세서와 비교해볼 때 명령어 형식의 차이가 있다. 그림 상단의 형식은 기존 MIPS16의 형식을 나타내고 하단의 형식이 새로 설계한 RISC의 명령어 형식이다.

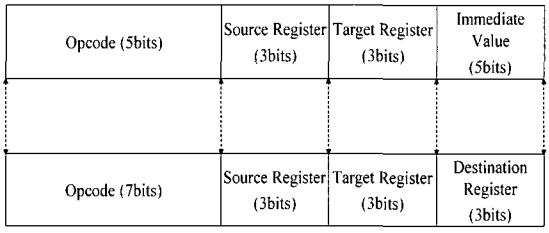

그림 1. 명령어 형식의 일치
Fig. 1. Mapping of Instructions format.

연산코드를 볼 때 기존의 MIPS16은 5bits를 이루고 있으나 설계한 명령어 형식은 7bits로 설계하였다. 또한 설계한 16-bit RISC Micro-processor는 기존 RISC 프로세서에 비해서 주소지정 방식을 단순화 하였다.

복잡도를 낮추면서도 기존의 명령어를 구현하기 위해서 모든 operand의 값을 레지스터로부터 읽어 들이며 Immediate 값은 사용하지 않았다. 또한 operand의 개수에 따라서 명령어를 구분하고 명령어에 따라서 사용되는 모듈을 선택되도록 하였다.

전체 명령어중에서 대부분은 Rs(Source register)와 Rt(Target register)의 연산에 의해 연산결과가 Rd (Destination register)에 기억된다. 또한 메모리 접근 명령어인 LW(Load Word), SW(Store Word)등의 명령어는 레지스터로부터 Offset과 Base주소를 가져와 연산을 하도록 설계되었다.

<표 1>에서는 구현된 전체 31개의 명령어 세트에 대한 형식과 설명을 나타내었다.

<표 2>에서는 Instruction Set Decoding Table을 나

표 1. 제안한 명령어 세트와 동작
Table 1. Proposed Instruction Set and Operation.

Mnemonic	Instruction
ADD	Add Word [ADD Rd, Rs, Rt]; [Rd ← Rs + Rt]
ADDU	Add Unsinged Word [ADDU Rd, Rs, Rt]; [Rd ← Rs+ Rt]
MULT	Multiply Word [MULT Rs, Rt]; [(LO, HI) ← Rs× Rt]
MULTU	Multiply Unsinged Word [MULTU Rs, Rt]; [(LO, HI) ← Rs × Rt]
SLT	Set on Less Then [SLT Rd, Rs, Rt]; [Rd ← (Rs < Rt)]
SLTU	Set on Less Then Unsinged [SLTU Rd, Rs, Rt]; [Rd ← (Rs < Rt)]
SUB	Subtract Word [SUB Rd, Rs, Rt]; [Rd ← Rs Rt]
SUBU	Subtract Unsigned Word [SUBU Rd, Rs, Rt]; [Rd ← Rs Rt]
BEQ	Branch on Equal ※ Rt=(Offset) [BEQ Rs, Rd, Offset]; [if Rs = Rd then Offset]
BGEZ	Branch on Greater Then or Equal to Zero [BGEZ Rs, Offset]; [if Rs ≥ 0 then Offset] ※Rt=(Offset)
BGTZ	Branch on Greater Then Zero ※ Rt=(Offset) [BGTZ Rs, Offset]; [if Rs > 0 then Offset]
BNE	Branch on Not Equal ※ Rt=(Offset) [BNE Rs, Rd, Offset]; [if Rs ≠ Rd then Offset]
JR	Jump Register [JR Rs]; [PC ← Rs]
NOP	No Operation
LW	Load Word ※ rs=(base), rt=(offset) [LW Rd, Offset(base)]; [Rd←memory(Offset+base)]
SW	Store Word ※ rs=(base), rt=(offset) [SW Rd, Offset(base)]; [memory(Offset+base)←Rd]
AND	And [AND Rd, Rs, Rt]; [Rd ← Rs AND Rt]
NOR	Not Or [NOR Rd, Rs, Rt]; [Rd ← Rs NOR Rt]
OR	Or [OR Rd, Rs, Rt]; [Rd ← Rs OR Rt]
XOR	Exclusive Or [XOR Rd, Rs, Rt]; [Rd ← Rs XOR Rt]
MOV	Move [MOV Rd, Rs]; [Rd ← Rs]
MFHI	Move From HI Register MFHI Rd]; [Rd ← HI]
MFLO	Move From LO Register [MFLO Rd]; [Rd ← LO]
MTHI	Move To HI Register [MTHI Rs]; [HI ← Rs]
MTLO	Move To LO Register [MTLO Rs]; [LO ← Rs]
BREAK	Breakpoint
SLL	Shift Word Left Logical ※ Rs=(sa) [SLL Rd, Rt, sa]; [Rd ← Rt << sa]
SRA	Shift Word Right Arithmetic ※ Rs=(sa) [SRA Rd, Rt, sa]; [Rd ← Rt >> sa] (Arithmetic)
ROL	Rotation Left Rs=(sa), Rt(MSB)=> Rt(LSB) [ROL Rd, Rt, sa]; [Rd ← Rt << sa]
ROR	Rotation Right Rs=(sa), Rt(LSB)=> Rt(MSB) [ROR Rd, Rt, sa]; [Rd ← Rt >> sa]
SRL	Shift Word Right Logical ※ Rs=(sa) [SRL Rd, Rt, sa]; [Rd ← Rt >> sa] (Logical)

표 2. 명령어 세트 디코딩
Table 2. Decoding of Instruction Set.

Op(15)=0			Op(11:9)							
			0	1	2	3	4	5	6	7
			000	001	010	011	100	101	110	111
Op(14:12)	0	000	SLL/NOP		SRL	SRA	ROL		ROR	
	1	001			MOV			LW		
	2	010	MFHI	MTHI	MFLO	MTLO				
	3	011	MULT	MLTU						
	4	100	ADD	ADDU	SUB	SUBU	AND	OR	XOR	NOR
	5	101			SLT	SLTU				
	6	110								
	7	111								

Op(15)=1			Op(11:9)							
			0	1	2	3	4	5	6	7
			000	001	010	011	100	101	110	111
Op(14:12)	0	000	JR	BGEZ			BEQ	BNE		BGTZ
	1	001								
	2	010	BREAK							
	3	011								
	4	100								
	5	101				SW				
	6	110								
	7	111								

타내어 명령어 세트가 연산코드에 따라서 구성된 것을 표시하였다.

Ⅲ. 프로세서 구조 및 설계

여러 종류의 단열회로 중에서 본 논문에서는 ECRL 형태를 사용하였으며, 전원부에는 AC형태의 전원클럭이 인가되어야 한다. 이를 이용하여 로직을 여러 단에 걸쳐 연결할 경우에는 90도의 위상차를 가지는 4개의 전원클럭에 의해서 연결되어야 한다. 따라서, 한 클럭동안에 4단의 연산이 순차적으로 수행된다. ECRL 단열회로는 구조상 차동신호를 사용한다.

마이크로프로세서는 Control Unit, PC, Barrel Shifter 그리고 ALU로 구성되며 본 논문에서 설계한 마이크로프로세서의 전체구조를 <그림 2>에 나타내었다.

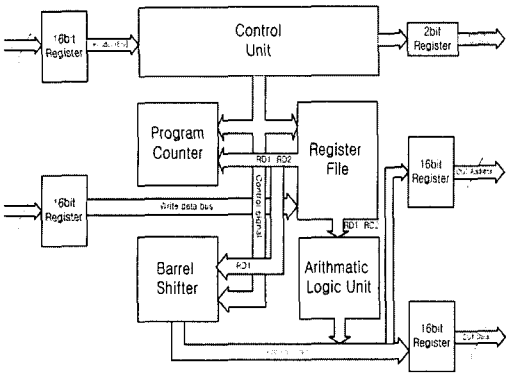

그림 2. 마이크로프로세서의 구조
Fig. 2. The structure of Microprocessor.

산술논리연산을 위한 ALU 블록은 덧셈, 뺄셈 그리고 논리연산을 담당한다. ALU는 뺄셈연산이 가능한 16-bit CLA, 16-bit 로직블록 그리고 MUX로 구성되어있다. ALU의 뺄셈연산은 '2'의 보수를 이용하므로 carry 입력신호가 필요하며, 총 7단을 거쳐서 연산결과를 출력한다. <그림 3>에 ALU와 Shifter의 블록 다이어그램을 나타내었다.

단열회로를 이용한 저전력 프로세서는 파이프라인 방식으로 동작을 하므로 이러한 동작방식에 적합한 Barrel Shifter를 설계하였다.

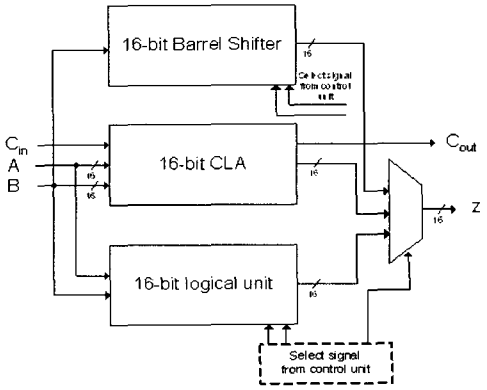

그림 3. ALU 와 Shifter의 구조
Fig. 3. The structure of ALU.

16-bit Barrel Shifter를 <그림 4>에 블록다이어그램으로 나타내었다. 이러한 구조는 전체 프로세서에서 위상을 줄이는데 이점이 있다.

레지스터 파일은 저장셀 array와 디코더 블록, sense amplifier 그리고 write circuit으로 구성되어져 있다. 본

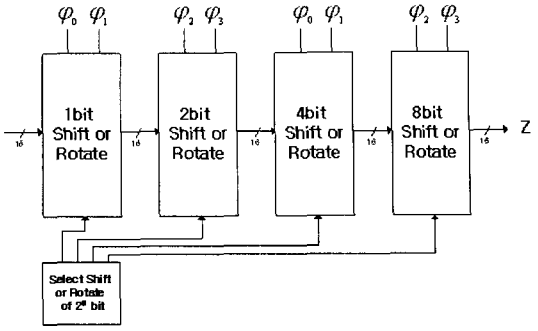

그림 4. 16-bit Barrel Shifter의 구조
Fig. 4. The Structure of 16-bit Barrel Shifter.

논문에서는 단열 프로세서에 사용하기 위해 3-read 1-write 다중포트 레지스터 파일을 설계하였다. 저장셀 array를 제외한 모든부분은 ECRL 형태로 설계되었다. 저장셀 array는 8주소의 저장셀을 가지며 일반적인 6-T SRAM 셀 구조로 설계하였다. 6-T Cell 구조는 면적의 효율성에서 단점이 있으나 동작속도가 빠르고 전력소모가 적은 장점이 있다. <그림 5>에서 16-bit 다중포트 레지스터 파일의 구조를 나타내었다.

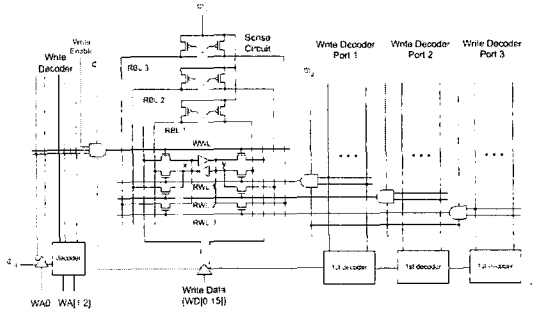

그림 5. 단열 레지스터파일의 구조
Fig. 5. The structure of adiabatic Register File.

쓰기동작은 3-phase를 사용하며 읽기 동작은 2-phase를 사용한다. 읽기 동작은 쓰기동작의 시작에서 1-phase의 지연을 갖고 시작되기 때문에 1개의 쓰기와 3개의 읽기 기능이 한 클럭 내에서 가능하다. 저장셀의 전원은 고정된 전원과 연결되며, 셀의 데이터가 변할 경우에만 에너지를 소모한다. Sense amplifier는 cross-coupled PMOS를 사용하였다.

Ⅳ. 전원클럭 발생기

ECRL 회로에서 에너지를 효율적으로 다시 전원으로 돌려주기 위해서는 AC형태의 전원공급이 필요하다. LC 공진회로는 4-phase의 전원클럭을 공급하기위해 사용되었다. <그림 6>에 전원클럭 발생기의 회로를 나타내었다.

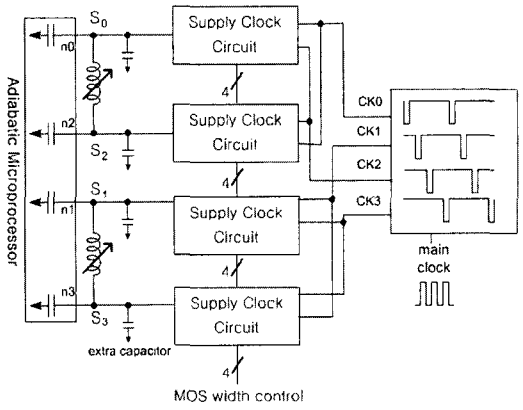

그림 6. 전원클럭 발생기의 구조
Fig. 6. The block diagram of supply clock generator.

본 논문에서 설계한 전원클럭 발생기는 1개의 인덕터, 두 개의 캐패시터 그리고 MOS 스위치로 구성되어 있다. <그림 7>에 전원공급 회로를 나타내었다.

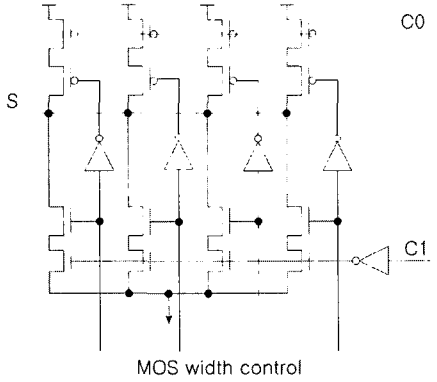

그림 7. 전원클럭 회로의 구조
Fig. 7. The schematic of supply clock circuit.

전원클럭 회로는 스위칭 MOS를 사용하여 단열회로 내부에 에너지를 공급하며, LC공진회로의 내부손실로 인한 진폭감소를 보상하여 항상 일정한 진폭을 유지하도록 동작한다.

마이크로프로세서의 동작주파수는 전원클럭 노드의 캐패시터의 값과 여기에 연결되는 인덕터 값의 곱에 의한 공진 주파수로 결정된다. 마이크로프로세서의 동작속도가 결정되면 공진 주파수 조정을 위해 인덕터의 값을 변경하여 동작시켰다. 전원클럭 노드의 등가 캐패시터 값은 식 (1)로 계산할 수 있다.

$$f=1/RC_{eq} \tag{1}$$

<표 3>에 마이크로프로세서의 전원클럭노드의 등가 캐패시터의 값을 나타내었다.

표 3. 전원클럭 노드의 등가 캐패시터값
Table 3. The equivalent capacitance of supply clock node.

node	S0	S1	S2	S3
C_{eq}[pF]	1.67	2.80	1.16	2.26

계산된 등가 캐패시터의 값들은 모두 다르기 때문에, 전원클럭 노드의 캐패시턴스 값을 동일하게 맞추어 주기 위해 별도의 캐패시터를 외부에 추가하여 준다.

Ⅴ. 실험결과

명령어 세트를 간소화 하여 Static CMOS 형태의 16-bit RISC Microprocessor를 설계하였다. 기본적인 구조는 저전력 단열 마이크로 프로세서와 동일하다. <그림 8>에 설계한 Static CMOS Microprocessor의 Modelsim 툴을 이용한 시뮬레이션 결과를 나타내었다.

①에서 in_sel=0, out_sel=0이므로 instruction memory의 명령어들을 fetch하고, out_data 값은 pc의 출력 값이 된다. ②에서 pc의 출력 값이 out_data로 출력됨을 보여주고 있다. Instruction memory의 0번 주소에 있는 값은 404a(hex)이므로 "add R1 R1 R2"를 의미하고 1번 레지스터에 있는 값을 두 번 더해 2번 레지스터에 저장하게 된다. 1번 레지스터에 있는 값은 1이므로 명령어 수행 후, 2번 레지스터에는 2가 저장된다.

<그림 9>에서는 adiabatic 프로세서를 Hynix 0.35μm full-custom 방식으로 설계한후 layout에서 net-list를 추출하여 SPICE 시뮬레이션을 수행한 결과를 나타내었다. Test Instruction code에 따라서 하위 9-bit의 두

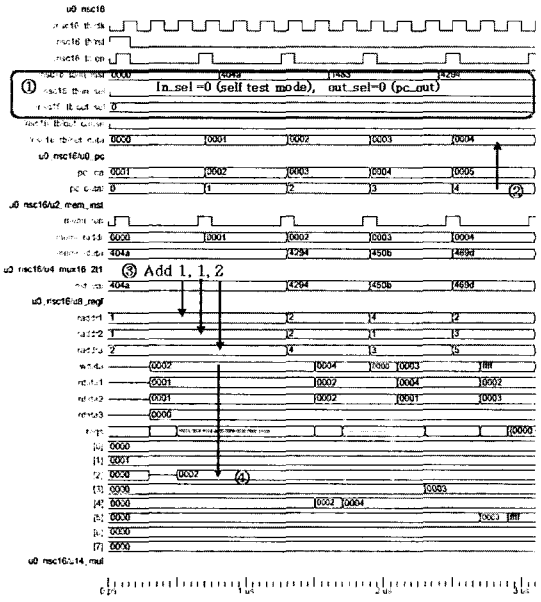

그림 8. 16-bit CMOS 마이크로프로세서의 시뮬레이션 결과
Fig. 8. Simulation result of 16-bit CMOS Microprocessor.

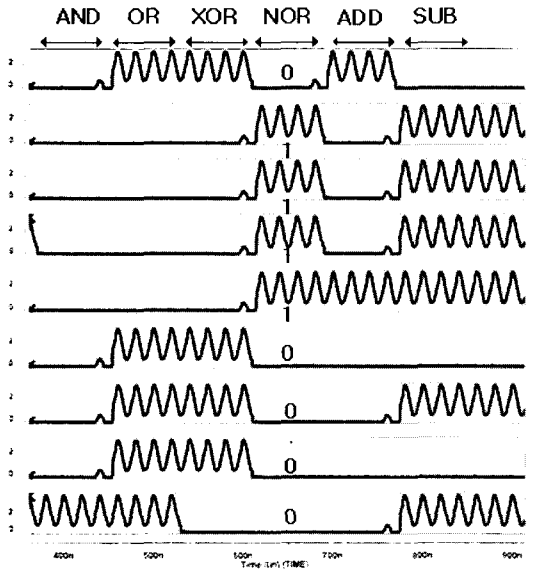

그림 9. 단열 마이크로프로세서의 시뮬레이션 결과
Fig. 9. Simulation result of adiabatic Microprocessor.

개의 레지스터 R1, R2의 연산결과를 보여준다.

레지스터 R1은 (0000 0100 0000 0011)2를 레지스터 R2는 (0000 0001 0000 1101)2의 값을 사용하여 산술과 논리연산을 보여준다. <그림 9>에 NOR 연산의 결과값을 표시하였다.

<그림 10>에서는 ECRL 인버터 회로의 에너지 소모를 그래프로 나타내었다.

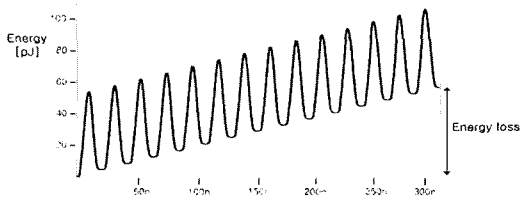

그림 10. ECRL 인버터의 에너지 복원 그래프
Fig. 10. Energy recovery plot of ECRL Inverter.

에너지 그래프를 통해서 전원클럭의 전압이 내려감에 따라 에너지가 복원되기 때문에 1-cycle 동안 소모하는 에너지가 적음을 알 수 있다.

<그림 11>은 설계된 단열 프로세서를 주파수에 따라서 에너지 소모를 측정하고, 단열 프로세서를 변경 설계한 CMOS 프로세서를 이용하여 DC전원을 인가하고 동일한 주파수에서 동일한 동작을 시켰을 경우 에너지를 비교측정한 그래프이다. 단열프로세서에서 소모한 에너지는 SPICE 모의실험에서 전압과 전류의 곱을 적분하는 회로를 이용하여 측정하였다. 에너지 비교는 다양한 주파수에서 비교하여 측정하였다. 동일한 명령어를 수행하였을 경우에 대해서 저전력 단열 마이크로프로세서가 CMOS 형태의 마이크로프로세서에 비해서 약 39%~41%의 에너지만을 소모한다.

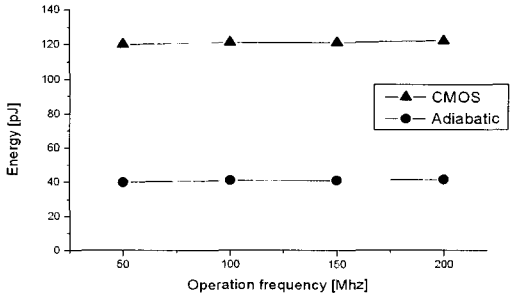

그림 11. 동작주파수에 따른 CMOS 마이크로프로세서와 저전력 단열 마이크로프로세서의 에너지
Fig. 11. Energy versus operation frequency for CMOS and adiabatic Microprocessor.

VI. Chip implementation

16-bit adiabatic 마이크로프로세서는 full-custom 방식으로 설계하였고, CMOS 마이크로프로세서는 HDL 설계 후 합성을 통하여 설계하였다. 두 칩의 레이아웃을 <그림 12>와 <그림 13>에서 나타내었다. Hynix 0.35μm 공정을 사용하여 설계된 칩의 core size는 adiabatic 프로세서의 경우 2.2mm × 1.3mm 이고, CMOS 프로세서의 경우 1.8mm × 1.8mm 이다.

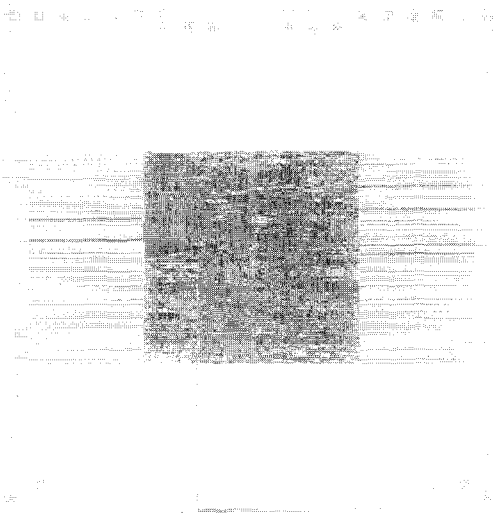

그림 12. CMOS 마이크로프로세서의 칩 레이아웃
Fig. 12. Chip layout of CMOS Microprocessor.

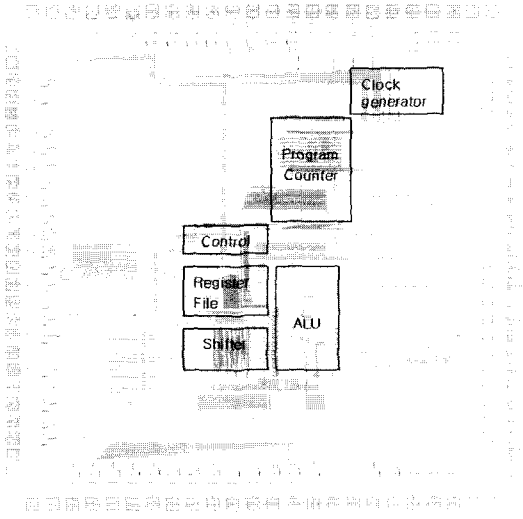

그림 13. 단열 마이크로프로세서의 칩 레이아웃
Fig. 13. Chip layout of adiabatic Microprocessor.

Fabrication을 위한 Die area는 4mm x 4mm를 사용하였고 100-pin QFP 패키지를 사용하여 칩을 제작하였으며, 현재 공정이 진행중이다.

VII. 결론 및 추후연구

0.35μm CMOS 공정을 이용하여 저전력 마이크로프로세서와 전원클럭 발생기를 설계하였고 비교실험을 위해 CMOS 마이크로프로세서도 설계하였다. 저전력 마이크로프로세서의 경우 레지스터파일의 저장셀을 제외한 모든 회로는 ECRL의 형태를 기반으로 하여 설계를 하였다. 단열 마이크로프로세서의 에너지 측정을 위해서 레이아웃에서 net-list를 추출하여 SPICE 시뮬레이션을 수행하였다. CMOS 마이크로프로세서는 HDL 시뮬레이터를 통해서 명령어의 동작을 확인하였다. 에너지 비교는 수정설계된 단열 프로세서에 DC전원을 인가하고 single-ended signaling을 이용하여 단열 프로세서와 비교하였다. SPICE를 이용한 모의실험에서 저전력 마이크로프로세서가 기존의 프로세서에 비해서 2.9~3.1배의 에너지 감소효과를 보였다. 본 논문에서 제안한 저전력 설계방식은 저전력을 요구하는 시스템이나 그밖의 디지털 회로에서 유용할 것으로 본다. 또한 고급기능을 갖는 프로세서와의 동등한 성능을 갖추면서 전력소모는 현저히 줄일 수 있는 프로세서에 대한 지속적인 연구가 필요할 것으로 생각된다. 본 연구에서 CAD Tool은 IDEC의 지원을 받았으며, 추후 칩에 대한 측정을 진행하고 Multiplier와 Cache memory를 추가한다면 고성능의 저전력 프로세서를 설계할 수 있다.

참 고 문 헌

[1] J. S. Denker, "A review of adiabatic computing," IEEE Symp. on Low Power Electronics, pp. 94~97, 1994.

[2] A. Kramer, J. S. Denker, S. C. Avery, A. G. Dickinson and T. R. Wik, "Adiabatic computing with the 2N-2N2D logic family," in Symp. on VLSI Circuits Dig. of Tech Papers, pp. 25~26, 1994.

[3] R. T. Hinman and M. F. Schlecht, "Power dissipation measurements on recovered energy

logic," in Symp. on VLSI Circuits, pp. 19~20, 1994.

[4] A. G. Dickinson and J. S. Denker, "Adiabatic dynamic logic," JSSC, vol. 30, pp. 311~315, 1995.

[5] C. W. Kim, S. M. Yoo and M. S. Kang, "Low-power adiabatic computing with NMOS energy recovery logic," electric letters, vol.36, pp. 1349~1350, Aug 2000.

[6] H. Mahmoodi-Meinnand, A. Afzali-Kusha and M. Nourani, "Adiabatic carry look-ahead adder with efficient power clock generator," IEEE Proc., vol.148, pp. 229~234, Oct 2001.

[7] L. Varga, F. Kovacs and G. Hosszu, "An efficient adiabatic charge-recovery logic," IEEE proc. southeastcon, pp.17~20, 2001.

[8] Y. Moon and D. K. Jeong, "An efficient charge recovery logic circuit," IEEE JSSC, vol.31, No. 4, pp. 514~522, Apr 1996.

[9] R. Brent and H.T. Kung, A regular layout for parallel adders, IEEE trans. on Computers, vol. C-31, no. 3, pp260~264, March 1982.

[10] Jan M, Rabaey "Digital Integrated Circuits" Prentice Hall Electronics and VLSI series pp. 359~362 pp.209.

[11] Joonho Lim, Kipaek kwon and Soo-Ik Chae, "Reversible energy recovery logic circuit without non-adiabatic energy loss" Electronics Letters, Vol:34 Issue:4, 19 Feb 1998.

[12] H. S. Lee, I .H. Na, H. H. Lee and Y. Moon "A 16-bit Adiabatic macro blocks with supply clock generator for micro-power RISC Data-path" ITC-CSCC2002, pp.1563~1566, July 2002.

저 자 소 개

申 暎 濬(學生會員)

2002년 : 숭실대학교 전자공학과 학사. 2002년~현재 : 숭실대학교 전자공학과 석사 재학. <주관심분야 : 저전력 프로세서 설계>

李 炳 勳(學生會員)

2003년 : 강남대학교 전자공학과 학사. 2003년~현재 : 숭실대학교 전자공학과 석사 재학. <주관심분야 : 저전력 프로세서 설계>

李 燦 豪(正會員)

1987년 2월 : 서울대학교 전자공학과 졸업(공학사). 1989년 2월 : 서울대학교 대학원 전자공학과 졸업(공학석사). 1994년 6월 : University of California, Los Angeles 전자공학과 졸업(공학 박사). 1994년 8월~1995년 2월 : 삼성전자 반도체연구소 선임연구원. 1995년 3월~현재 : 숭실대학교 정보통신전자공학부 부교수. <주관심분야 : 채널 코덱의 VLSI 구현, SoC on-chip-network, SoC 설계 방법론, 저전력 프로세서 설계 및 암호프로세서 설계 등임>

文 龍(正會員)

1990년 : 서울대학교 전자공학과 학사. 1992년 : 서울대학교 전자공학과 석사. 1997년 : 서울대학교 전자공학과 박사. 1997년~1999년 : LG 반도체 선임연구원. 1999년 9월~현재 : 숭실대학교 정보통신전자공학부 조교수. <주관심분야 : 저전력 회로설계, 혼성신호 IC, RF IC 및 고속통신시스템 등임>