유전자 알고리즘 연산 명령어를 포함한 마이크로프로세서의 설계와 구현

최 정 필, 이 재 진, 송 기 용
{welcomerain, ceicarus}@archi.chungbuk.ac.kr gysong@chungbuk.ac.kr
충북대학교 컴퓨터공학과
충북 청주시 흥덕구 개신동 산48번지
Tel +82-43-261-2452, Fax +82-43-262-2449

키워드: 유전자 알고리즘, 마이크로프로세서, VHDL, 교배, 돌연변이

요 약

본 논문에서는 VHDL을 이용하여 교배, 돌연변이와 같은 유전자 알고리즘의 연산 명령어를 포함하고, 4가지의 주소 지정 방식을 가지며 32개의 명령어를 수행하는 마이크로프로세서를 설계하였다. 설계한 마이크로프로세서는 16-bit 주소로 64Kbyte의 메모리를 참조하고, 직·병렬 입출력 포트와 IR, PC, SP, Y, BASE, MA, MD, AC, PIN, SIN, POUT, SOUT의 내부 레지스터를 가진다. 설계한 마이크로프로세서를 시뮬레이션 통해 검증하였고 FPGA 칩상에 합성하였다.

1. 서 론

유전자 알고리즘(GA: Genetic Algorithm)은 다윈의 진화론을 기반으로 하여 개발된 성능이 탁월하며, 도메인에 독립적인 탐색 기법이다. Solution space에서 최적해로 접근하기 위해서 효율적이고 규칙적인 탐색 방법을 필요로 하게 되는데, 이를 위해 자연적인 진화의 과정을 모방하였다. 어떤 특성을 갖는 개체(individual)가 살아남을 가능성이 더 크다는 가정에 의해 자연적 선택 이론(natural selection theory)을 기반으로 하고 있다[1].

모든 유전자 알고리즘은 해집단(population)에서 동작한다. 해집단의 각 개체는 염색체(chromosome)로 불리우며 이러한 개체들은 이진 문자열(binary strings)로 만들어질 수 있다. 유전자 알고리즘의 각 반복에서는 더 좋은 해(solution)를 생성하기 위해서 현재 존재하는 해집단으로부터 새로운 새대(generation)를 만드는 진화의 과정을 거치게 된다.

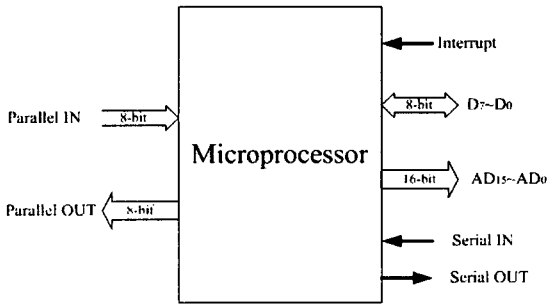
교배(crossover)는 재생성(reproduction)을 위해 사용되는 대표적인 연산이다. 양쪽 부모(parents)로부터 일부분(portion)을 받아 조합(combine)함으로써 자식(offspring)이라고 하는 새로운 개체를 만든다. 자식들은 부모의 일부분으로 만들어지기 때문에 부모의 특징을 그대로 상속받게 된다. 돌연변이(mutation)는 해집단의 각 구성원으로부터 아주 작은 확률로 증가되는 변화(incremental change)를 말한다. 돌연변이에 의해 해집단은 새로운 특성을 가질 수 있다[2].

본 논문에서는 교배, 돌연변이와 같은 유전자 알고리즘 연산 명령어를 포함하고, 즉치 주소(Immediate Addressing), 직접 주소(Direct Addressing), 레지스터 간접 주소(Register Indirect Addressing), 레지스터 베이스 주소(Register Based Addressing)의 4가지 주소 지정 방식을 가지며 32개의 명령어와 인터럽트 처리를 수행하는 마이크로프로세서를 VHDL로 설계하고 Active-HDL을 이용한 시뮬레이션을 통하여 검증한 후 Synopsys를 이용하여 합성하고 FPGA 상에 구현하였다.

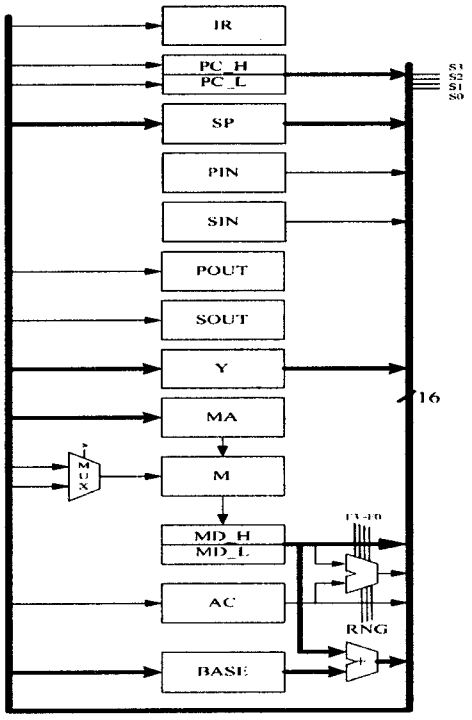
2. 유전자 알고리즘 연산 명령어를 포함하는 프로세서의 구성

2.1 프로세서의 기본 구성

본 논문에서 설계한 마이크로프로세서는 1Kbyte의 마이크로 프로그램과 64Kbyte의 데이터 메모리 주소 공간, 그리고 IR, PC, SP, Y, BASE, MA, MD, AC, PIN, SIN, POUT, SOUT의 내부 레지스터를 가진다. IR, AC, PIN, SIN, POUT, SOUT 레지스터의 크기는 8-bit로 되어있으며, PC, SP, Y, BASE, MA, MD 레지스터는 각 16-bit로 구성하였다. [그림-1]은 본 논문에서 구현한 마이크로프로세서의 외부 입출력을 나타내며, [그림-2]는 마이크로프로세서의 내부 구성도이다[3~6].



[그림-1] 마이크로프로세서의 외부 입출력



[그림-2] 마이크로프로세서의 내부 구성도

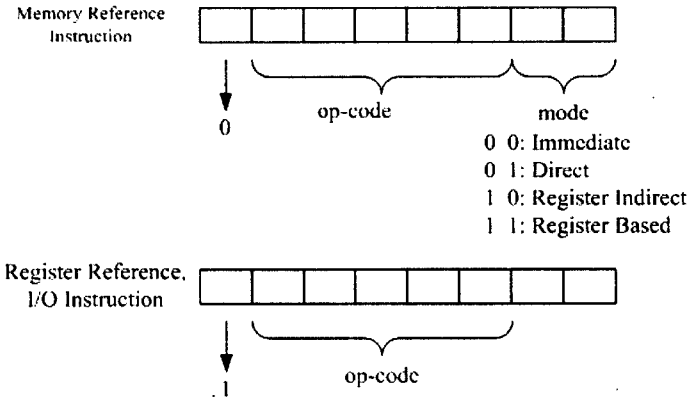
설계된 프로세서가 사용하는 레지스터는 총 12개이며, 각 레지스터의 기능을 <표 1>에 나타내었다. 각 레지스터는 고유한 기능을 수행하며, 레지스터 데이터는 공용 버스를 이용해 전달되거나 외부와 입출력된다.

[표-1] 레지스터의 기능

레지스터	기능
IR	현재의 동작 코드를 지정한다.
PC	다음 수행 될 명령의 번지를 지정한다.
SP	메모리 스택의 주소를 지정한다.
Y	간접 주소 지정방식에 대한 주소를 저장한다.
MA	메모리 주소를 지정한다.
MD	메모리 데이터를 저장한다.
AC	연산의 결과나 입출력의 데이터를 저장한다.
PIN	외부로부터 데이터를 입력받는 병렬 포트
SIN	외부로부터 데이터를 입력받는 직렬 포트
POUT	외부로 데이터를 출력하는 병렬 포트
SOUT	외부로 데이터를 출력하는 직렬 포트
BASE	기준이 되는 주소를 저장한다.

2.2 명령어의 구성

마이크로프로세서의 명령어 OP Code는 <그림 3>과 같은 구조로 구성되어 있다. 명령어는 크게 메모리 참조 명령과 레지스터, I/O 참조 명령으로 구분된다.



[그림-3] 마이크로 프로세서의 명령어

메모리 참조 명령어와 레지스터, I/O 참조 명령어는 최상위 비트로서 구분을 한다. 즉 최상위 비트 IR(7)이 '0'이면 메모리 참조 명령어, '1'이면 레지스터 또는 I/O 참조 명령어로 구분된다.
메모리 참조 명령어와 레지스터, I/O 참조 명령어 모두 IR(6-2) 비트로 명령어를 나타내기 때문에 디코딩 과정이 필요하다. 또한 IR(1-0) 비트로 즉치 주소, 직접 주소, 레지스터 간접 주소, 레지스터 베이스 주소 4가지 방식을 구분한다.

2.3 명령어의 종류

마이크로프로세서의 명령어는 32가지의 종류로 정의한다. [표-2]와 [표-3]은 각 메모리 참조 명령어와 레지스터, I/O 참조 명령어들을 나타낸 것이다.

[표-2] 메모리 참조 명령어

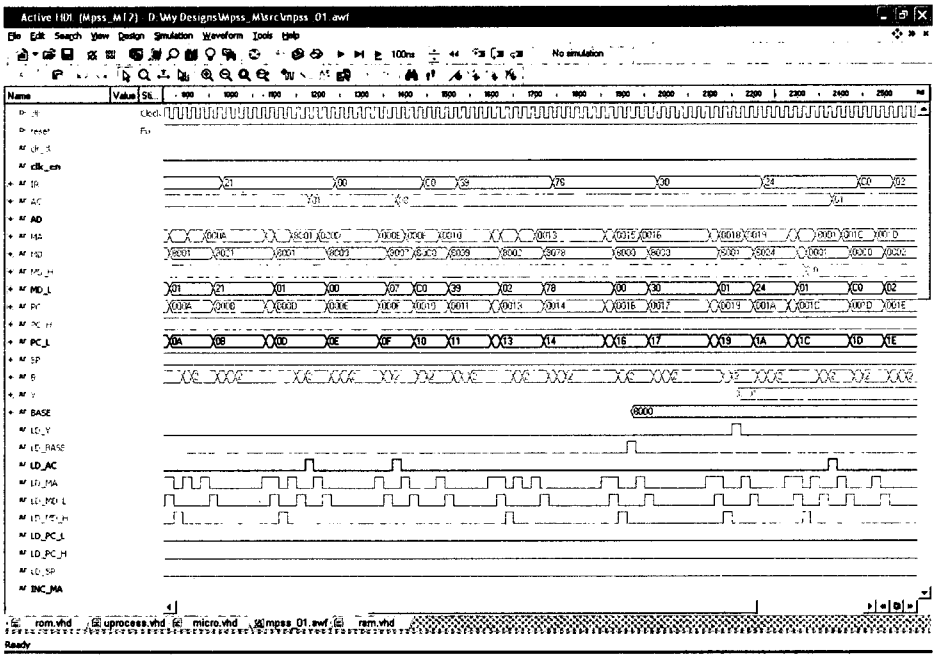
명령어	code	Description
ADD	00000	AC←AC+[M]/imm; CF,ZF
AND	00001	AC←AC · [M]/imm; ZF
OR	00010	AC←AC∨[M]/imm; ZF
CMP	00011	CF←AC-[M]/imm; CF,ZF
LDA	00100	AC←[M]/imm
LDSP	00101	SP←[M]/imm
LDY	00110	Y←[M]/imm
ST	00111	[M]←AC
BR	01000	PC←imm
BC	01001	if CF='1' then PC←imm
BZ	01010	if ZF='1' then PC←imm
BY	01011	if YZF='1' then PC←imm
CALL	01100	[SP]←PC, PC←imm, SP←SP+1
RET	01110	PC←[SP-1]
XOR	10000	AC←AC⊕[M]/imm; ZF
LDB	10001	BASE←[M]/imm
XOVRML	10010	AC←AC(7..i)&[M](i-1..0)

[표-3] 레지스터, I/O 참조 명령어

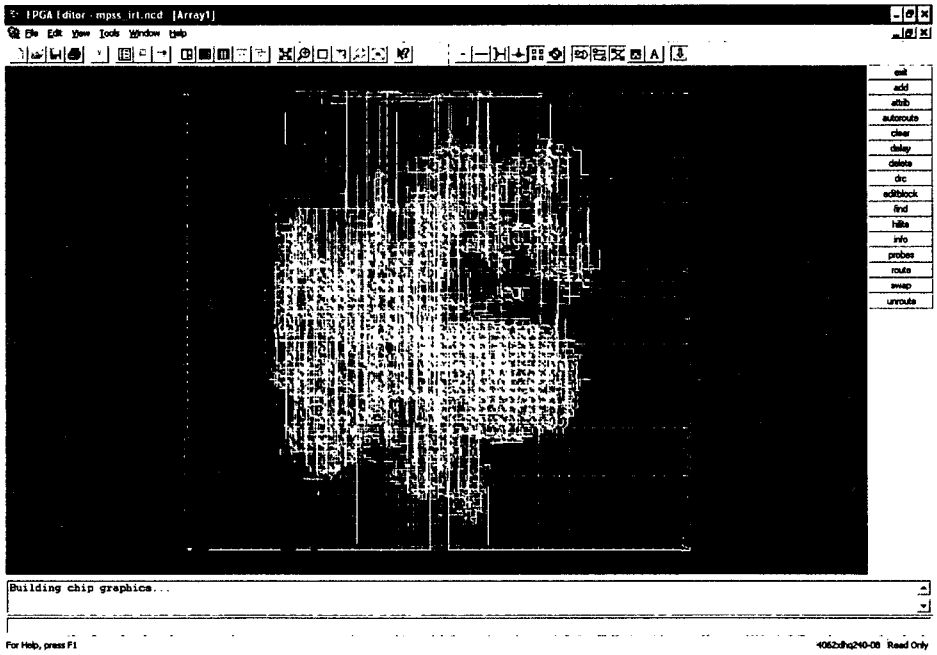
명령어	code	Description
INC	00000	Y←Y+1
COM	00001	AC←AC'
SHL	00010	AC←SHL(AC)
SHR	00011	AC←SHR(AC)
ROTL	00100	AC←ROTL(AC)
ROTR	00101	AC←ROTR(AC)
PIN	00110	AC←PIN
SIN	00111	AC←SIN
POUT	01000	POUT←AC
SOUT	01001	SOUT←AC
CC	01010	CF←0
HALT	01011	SS←0
MUT1	01101	if AC(i)='1' then AC(i)='0' else AC(i)='1'

3. 시뮬레이션 및 합성

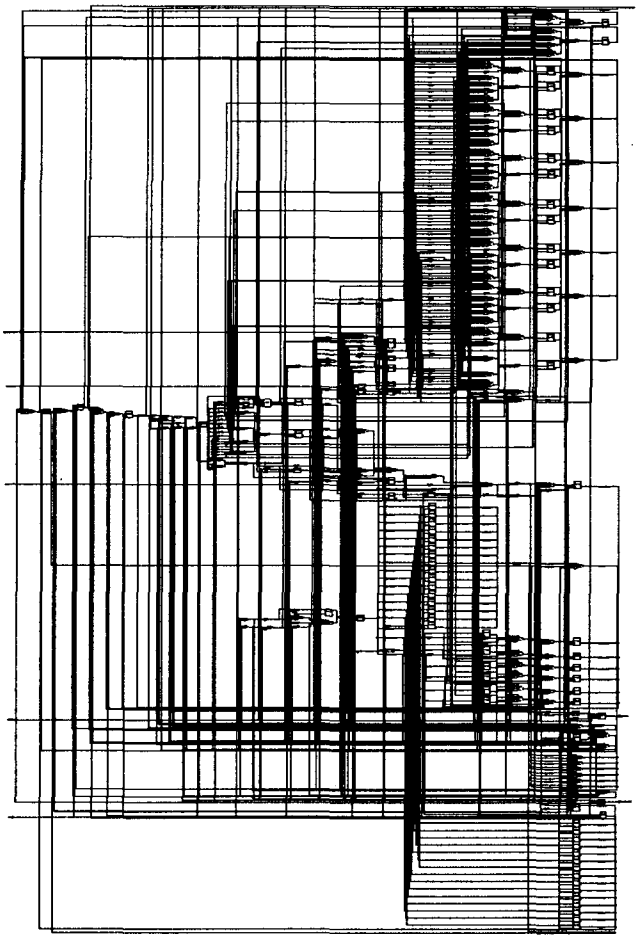
설계된 프로세서를 Active-HDL을 사용하여 시뮬레이션 한 결과 [그림-4]과 같이 정상적으로 각 명령어를 수행하였다. 이것을 Xilinx Foundation을 사용하여 합성한 후 Xilinx FPGA 상에 구현하였다[그림-5]. 또한 Synopsys를 이용한 합성 결과인 마이크로프로세서의 전체구성을 [그림-6]에 보였다.



[그림-4] 시뮬레이션 결과



[그림-5] 설계된 프로세서의 FPGA 구현



[그림-6] Synopsys를 이용한 합성 결과

4. 결론

본 논문에서는 교배, 돌연변이와 같은 유전자 알고리즘 연산 명령어를 포함하고, 즉치 주소 방식, 직접 주소 방식, 레지스터 간접 주소 방식, 레지스터 베이스 주소 방식의 4가지 주소 지정 방식을 가지고 32개의 명령어와 인터럽트 처리를 수행하는 마이크로프로세서를 VHDL로 설계하고 Active-HDL을 사용하여 시뮬레이션 한 결과 정상적인 동작을 하는 프로세서임을 확인 하였다.

또한 Xilinx FPGA VL-XC4062XL 상에 합성한 결과 ROM에 저장한 테스트 프로그램을 정확히 수행함을 알 수 있었다.

참고문헌

[1] Sadiq M. Sait, *Iterative computer Algorithms with Applications in Engineering*, IEEE Computer Society.

[2] Pinaki Mazumder, *Genetic Algorithms for VLSI Design, Layout & Test Automation*, Prentice-Hall, 1999

[3] Douglas J. Smith, *HDL Chip Design*, Doone Publications, 2000

[4] Charles H. Roth, Jr., *Digital Systems Design using VHDL*, PWS Publications, 1998

[5] Hohn P.Hayes, Computer Architecture and Organization, McGraw-Hill.

[6] 박세현, *디지털 컴퓨터 설계와 구현*, 그린

[7] Xilinx, 1999 *Xilinx Data Book.*