



(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl.

G06F 9/38 (2006.01)

G06F 9/00 (2006.01)

G06F 13/00 (2006.01)

(45) 공고일자 2007년06월25일

(11) 등록번호 10-0731976

(24) 등록일자 2007년06월19일

(21) 출원번호 10-2005-0057530

(22) 출원일자 2005년06월30일

심사청구일자 2005년06월30일

(65) 공개번호 10-2007-0002163

(43) 공개일자 2007년01월05일

(73) 특허권자 전자부품연구원
경기도 성남시 분당구 야탑동 68번지

(72) 발명자 이승은
경기 성남시 분당구 야탑동 68

정용무
경기 성남시 분당구 야탑동 68

최종찬
경기 성남시 분당구 야탑동 68

(74) 대리인 특허법인지명

(56) 선행기술조사문헌
KR1020040041650 A
US5966534 A
WO03025784 A2

US20020002573 A1
US5970254 A
US 6,247,036

심사관 : 노지명

전체 청구항 수 : 총 12 항

(54) 재구성 가능 프로세서의 효율적인 재구성 방법

(57) 요약

본 발명은 재구성 가능 프로세서의 효율적인 재구성 방법에 관한 것으로서, 응용 프로그램에 사용된 명령어를 분석하여, 재구성 가능한 N 개의 명령어 중에서 M 개(M<N)의 재구성 모드를 선택하고 상기 재구성 모드를 상기 재구성 제어 유닛에 설정하는 선 재구성(Pre-Configuration) 단계와; 응용 프로그램 실행시에, 상기 M 개의 재구성 모드를 식별하기 위해 각 명령어에 추가되는 재구성 모드 비트를 해석하여, 당해 명령어에 대응하는 데이터 경로를 실시간으로 재구성하는 실시간 재구성 단계를 포함한다. 본 발명에 따르면, 응용 프로그램에서 요구되는 명령어를 기반으로 프로세서의 재구성 모드를 먼저 설정하는 선 재구성을 수행함으로써, 실제 응용 프로그램이 수행될 때의 실시간 재구성시에 먼저 결정된 모드만을 이용하여 재구성에 요구되는 코드의 사이즈 및 하드웨어를 경량화시킬 수 있다.

대표도

도 3

특허청구의 범위

청구항 1.

명령어에 추가된 재구성 모드 비트에 따라 데이터 경로를 재구성하는 재구성 제어 유닛이 구비된 재구성 가능 프로세서에서 데이터 경로의 재구성 방법으로서,

응용 프로그램에 사용된 명령어를 분석하여, 재구성 가능한 N 개의 명령어 중에서 M 개($M < N$)의 재구성 모드를 선택하고, 상기 재구성 모드를 상기 재구성 제어 유닛에 설정하는 선 재구성(Pre-Configuration) 단계와,

응용 프로그램 실행시에, 상기 M 개의 재구성 모드를 식별하기 위해 각 명령어에 추가되는 재구성 모드 비트를 해석하여 당해 명령어에 대응하는 데이터 경로를 실시간으로 재구성하며, 상기 재구성 모드로 선택되지 아니한 명령어는 기본 데이터 경로를 사용하는 실시간 재구성 단계

를 포함하고,

상기 선 재구성 단계는 상기 응용 프로그램에 사용된 각 명령어의 빈도수를 비교하여 상기 M 개의 재구성 모드를 선택하는 것

인 재구성 가능 프로세서의 데이터 경로 재구성 방법.

청구항 2.

삭제

청구항 3.

제1항에 있어서, 상기 선 재구성 단계는

상기 응용 프로그램에 사용된 각 명령어가 기본 데이터 경로를 이용할 때 요구하는 사이클 수를 비교하여 상기 M 개의 재구성 모드를 선택하는 것인 재구성 가능 프로세서의 데이터 경로 재구성 방법.

청구항 4.

제1항에 있어서, 상기 선 재구성 단계는

상기 응용 프로그램을 수행함에 있어서 각 명령어가 요구하는 총 사이클 수를 비교하여, 상기 M 개의 재구성 모드를 선택하는 것인 재구성 가능 프로세서의 데이터 경로 재구성 방법.

청구항 5.

제1항에 있어서, 상기 선 재구성 단계는

각 명령어에 대하여 재구성 시에 감소되는 총 사이클 수를 비교하여, 상기 M 개의 재구성 모드를 선택하는 것인 재구성 가능 프로세서의 데이터 경로 재구성 방법.

청구항 6.

제1항 내지 제5항 중 어느 하나의 항에 있어서,

상기 선 재구성 단계에서 재구성 모드로 선택되지 아니한 재구성 가능한 명령어는 상기 재구성 모드 비트가 기본 데이터 경로를 사용하는 기본 모드로 지정되는 것

인 재구성 가능 프로세서의 데이터 경로 재구성 방법.

청구항 7.

제6항에 있어서,

상기 명령어에 추가되는 재구성 모드 비트는 $\log_2 M$ 개인 재구성 가능 프로세서의 데이터 경로 재구성 방법.

청구항 8.

명령어에 추가된 재구성 모드 비트에 따라 데이터 경로를 재구성하는 재구성 제어 유닛이 구비된 재구성 가능 프로세서에서, 상기 재구성 제어 유닛은,

응용 프로그램에 사용된 N 개의 명령어 중에서 선택된 M 개($M < N$)의 재구성 모드를 저장하며,

응용 프로그램 실행시에, 상기 M 개의 재구성 모드를 식별하기 위해 각 명령어에 추가되는 재구성 모드 비트를 해석하여, 당해 명령어에 대응하는 데이터 경로를 실시간으로 재구성하며, 상기 재구성 모드로 선택되지 아니한 명령어는 기본 데이터 경로를 사용하며,

상기 응용 프로그램에 사용된 각 명령어의 빈도수를 비교하여 상기 M 개의 재구성 모드를 선택하는 것

인 재구성 가능 프로세서.

청구항 9.

삭제

청구항 10.

제8항에 있어서, 상기 재구성 제어 유닛은

상기 응용 프로그램에 사용된 각 명령어가 기본 데이터 경로를 이용할 때 요구하는 사이클 수를 비교하여 상기 M 개의 재구성 모드를 선택하는 것인 재구성 가능 프로세서.

청구항 11.

제8항에 있어서, 상기 재구성 제어 유닛은

상기 응용 프로그램을 수행함에 있어서 각 명령어가 요구하는 총 사이클 수를 비교하여, 상기 M 개의 재구성 모드를 선택하는 것인 재구성 가능 프로세서.

청구항 12.

제8항에 있어서, 상기 재구성 제어 유닛은

각 명령어에 대하여 재구성 시에 감소되는 총 사이클 수를 비교하여, 상기 M 개의 재구성 모드를 선택하는 것인 재구성 가능 프로세서.

청구항 13.

제8항 내지 제12항 중 어느 하나의 항에 있어서,

상기 재구성 모드로 선택되지 아니한 명령어는 상기 재구성 모드 비트가 기본 데이터 경로를 사용하는 기본 모드로 지정되는 것인 재구성 가능 프로세서.

청구항 14.

제13항에 있어서,

상기 명령어에 추가되는 재구성 모드 비트는 $\log_2 M$ 개인 재구성 가능 프로세서.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 재구성 가능 프로세서에 관한 것으로서, 보다 구체적으로는 재구성 가능 프로세서의 재구성에 있어 응용 프로그램에서 요구되는 명령어(instruction)를 기반으로 프로세서의 재구성 모드를 먼저 설정하는 선재구성(Pre-Configuration)을 수행함으로써, 실제 응용 프로그램이 수행될 때의 실시간 재구성시에 사전에 결정된 모드를 이용하여 재구성에 요구되는 코드의 사이즈 및 하드웨어를 경량화할 뿐만 아니라, 이와 같이 모드를 미리 설정함으로써 응용 프로그램의 특징에 맞게 재구성하여 사용할 수 있는 유연성을 부여할 수 있도록 하는 재구성 가능 신호처리기의 효율적인 재구성 방법에 관한 것이다.

일반적으로, 신호처리는 신호처리 및 연산을 위하여 데이터 경로(Data Path)를 포함하고 있으며, 연산은 이러한 데이터 경로를 통해 이루어진다.

도 1은 통상의 병렬 처리 디지털 신호처리(DSP)의 구성을 예시한 것으로서, 도시된 바와 같이 프로그램 메모리(도시되지 않음)로부터 프로그램 버스(Program Bus)(110)를 경유하여 명령어(Instruction)를 수신하고 해당 명령어에 따른 연산을 수행하도록 제어하는 제어 유닛(Control Unit)(120)과, 상기 제어 유닛에 의하여 해독된 명령어에 따라 주소 연산을 수행하는 주소 생성 유닛(Address Generation Unit)(130)과, 상기 제어 유닛에 의하여 해독된 명령어에 따라 데이터 연산을 수행하는 데이터 연산 유닛(Data Arithmetic Unit)(140)과, 상기 제어 유닛과 주소 생성 유닛 및 데이터 연산 유닛 간의 데이터 교환을 위한 데이터 버스(Data Bus)(150)로 구성되어 있다.

데이터 연산과 관련하여, 데이터 연산 유닛(140)은 m 개의 데이터 경로 유닛(Data Path Unit)(142_1, 142_2, ..., 142_m)을 서브유닛으로서 구비하여 최대 m 개의 로직 연산을 수행할 수 있다. 또한, 데이터 연산 유닛(140)은 데이터 연산에 필요한 데이터 및 그 연산 결과를 저장하는 레지스터 파일(144)을 추가로 구비하며, 각 데이터 경로 유닛(142_1, 142_2, ..., 142_m)은 레지스터 파일(144)에 저장된 데이터를 이용하여 특정 연산을 수행하도록 데이터 연산 유닛(140)의 내부에 마련된 논리 연산 유닛 사이에 데이터 경로를 구성한다.

이와 같은 DSP의 동작을 살펴보면, 먼저 프로그램 메모리로부터 읽은 명령어를 명령어 해독기(126)가 해독하고, 내부 레지스터 파일(144)은 데이터 메모리(도시되지 않음)로부터 해당 데이터를 로딩한다. 연산은 레지스터 파일(144)의 데이터를 기반으로 데이터 경로 유닛 내부의 데이터 경로(Data Path)를 따라 수행된다.

그런데, 데이터 경로가 고정된 신호처리기 또는 프로세서는 하나의 클럭 사이클 동안에 데이터 경로를 따라 미리 지정된 유형의 연산만을 수행할 수 있기 때문에, 미리 지정된 연산 이외의 연산은 클럭 사이클을 반복하여야 하는 단점이 있다. 이에 본 발명의 발명자는 2005년 5월 27일 출원된 한국특허출원 제2005-45063호 "데이터 경로의 재구성이 가능한 디지털 신호 처리기"를 제안한 바 있다.

도 2는 전술한 한국특허출원 제2005-45063호에 개시된 디지털 신호 처리기 내부의 데이터 경로 유닛을 도시한 것이며, 단위 연산을 수행하는 복수개의 산술 논리 유닛(210_1, 210_2, ..., 210_n)과, 입력 데이터 및 상기 복수개의 산술 논리 유닛의 출력 데이터 중에서 각 산술 논리 유닛에 입력되는 데이터를 선택하여 출력하는 한 쌍의 입력 멀티플렉서(220_1, 220_2, ..., 220_n)와, 상기 복수개의 산술 논리 유닛의 출력 데이터 중에서 하나를 선택하여 출력하는 출력 멀티플렉서(230)와, 상기 출력 멀티플렉서 및 입력 멀티플렉서의 데이터 선택을 제어하는 재구성 제어 유닛(Reconfiguration Control Unit)(240)으로 이루어져 있다.

상기 복수개의 산술 논리 유닛(210_1 내지 210_n)으로는 단위 연산을 수행하며, 예컨대, 곱셈, 시프트(Barrel Shifter), 덧셈, 비교, 최대/최소 연산, 논리 연산(AND/OR/XOR/NOT) 및 라운드/포화 연산기(Round/Saturation) 등을 각각 수행할 수 있다. 그리고, 이들 산술 논리 유닛(210_1 내지 210_n)의 입력단에는 한 쌍의 입력 멀티플렉서(220_1, 220_2, ..., 220_n)가 상호 대응하도록 연결되어 있다.

한 쌍의 입력 멀티플렉서(220_1, 220_2, ..., 220_n)에는 기 설명한 레지스터 파일(도시되지 않음)로부터 액세스되는 예컨대 세 개의 입력 데이터(Input #1, Input #2, Input #3)와 산술 논리 유닛(210_1 내지 210_n)으로부터 출력되는 데이터가 각각 입력되며, 재구성 제어 유닛(Reconfigure Control Unit)(240)으로부터 입력되는 제어 신호에 따라 이들 중에서 하나의 데이터를 각각 선택될 수 있다.

출력 멀티플렉서(230)는 재구성 제어 유닛(240)으로부터 입력되는 제어 신호에 따라, 상기 복수개의 산술 논리 유닛(210_1 내지 210_n)으로부터 출력되는 데이터 중에서 하나를 선택한다.

재구성 제어 유닛(240)은 전술한 출력 멀티플렉서 및 입력 멀티플렉서의 데이터 선택을 동시에 일괄적으로 제어함으로써, 특정 산술 논리 유닛에 의해 수행된 연산 결과가 다른 산술 논리 유닛의 입력으로 사용될 수 있도록 하며, 최종적으로 연산이 완료된 결과를 선택하여 출력하도록 구성된다. 산술 논리 유닛(210_1)에 대응하는 한 쌍의 입력 멀티플렉서(220_1)의 경우를 예로서 설명하면, 하나의 명령어 워드에 의하여 연산이 지시된 세 개의 입력 데이터(Input #1, Input #2, Input #3) 중에서 두 개의 데이터를 선택하여 산술 논리 유닛(210_1)에 입력할 수 있으며, 이로부터 출력되는 데이터는 다시 각 입력 멀티플렉서에 입력됨으로써 타 연산을 연속하여 수행할 수 있다.

이와 같이 데이터 경로를 재구성 하는 재구성 가능 프로세서의 경우 재구성 제어 유닛의 명령에 따라 데이터 경로가 재구성된다. 즉, 재구성 제어 유닛은 각 로직 연산 유닛의 입출력을 선택하는 멀티플렉서 선택 신호를 생성함으로써, 전체 데이터 경로를 재구성하는 것이다.

그런데, 이러한 재구성 제어 유닛의 동작과 관련하여, 실시간 재구성을 위한 재구성 관련 정보를 명령어에 직접 포함시키거나, 재구성 관련 회로를 내장시키는 방법으로 재구성 관련 명령을 생성할 수 있다.

먼저, 프로세서의 명령어가 재구성에 요구되는 신호를 모두 포함하는 경우에, 재구성 제어 유닛은 이러한 명령어를 해석하여 각 멀티플렉서 신호를 생성할 수 있다. 이는 N개의 재구성 가능한 로직 연산 유닛을 포함하는 재구성 가능 프로세서에 있어서, $\log_2 N$ 비트의 재구성 관련 정보를 포함하는 헤더를 각 명령어에 추가해야 하는 단점이 있다.

다음으로, 재구성 제어 유닛이 자체적으로 재구성을 위한 멀티플렉서 신호를 생성하는 방법이 있다. 즉, 자체적인 하드웨어 로직을 가지고 각각의 제어 신호를 생성하는 것이다. 이는 명령어 길이에 영향을 미치지 않으나, 추가적인 하드웨어를 내장해야 하며, 한번 결정된 재구성 방법을 응용 제품에 따라 변경할 수 없는 단점이 있다.

이와 같이, 종래의 재구성 가능 프로세서의 재구성 방법은 전체 명령어의 길이가 길어지거나, 다양한 응용에 사용하기 위한 재구성의 유연성이 떨어지는 단점이 있다.

발명이 이루고자 하는 기술적 과제

전술한 문제점을 해결하고자, 본 발명은 재구성 가능 프로세서의 재구성에 있어 재구성에 요구되는 코드의 사이즈 및 하드웨어를 경량화함과 아울러, 응용 프로그램의 특성에 맞게 프로세서를 재구성하여 사용할 수 있도록 하는 데 그 목적이 있다.

발명의 구성

전술한 목적을 달성하기 위하여, 본 발명의 제1 측면에 따르면, 명령어에 추가된 재구성 모드 비트에 따라 데이터 경로를 재구성하는 재구성 제어 유닛이 구비된 재구성 가능 프로세서에서 데이터 경로의 재구성 방법이 제공된다. 그리고, 상기 재구성 방법은 응용 프로그램에 사용된 명령어를 분석하여, 재구성 가능한 N 개의 명령어 중에서 M 개($M < N$)의 재구성 모드를 선택하고, 상기 재구성 모드를 상기 재구성 제어 유닛에 설정하는 선 재구성(Pre-Configuration) 단계와; 응용 프로그램 실행시에, 상기 M 개의 재구성 모드를 식별하기 위해 각 명령어에 추가되는 재구성 모드 비트를 해석하여 당해 명령어에 대응하는 데이터 경로를 실시간으로 재구성하며 상기 재구성 모드로 선택되지 아니한 명령어는 기본 데이터 경로를 사용하는 실시간 재구성 단계를 포함한다.

이 때, 상기 선 재구성 단계는 상기 응용 프로그램에 사용된 각 명령어의 빈도수, 각 명령어가 기본 데이터 경로를 사용할 때 요구하는 사이클 수, 각 명령어가 응용 프로그램 전체에서 요구하는 총 사이클 수, 각 명령어에 대하여 재구성 시에 감소되는 총 사이클 수 등을 비교하여 상기 M 개의 재구성 모드를 선택할 수 있다.

더욱 바람직하게는, 상기 선 재구성 단계에서 재구성 모드로 선택되지 아니한 재구성 가능한 명령어는 상기 재구성 모드 비트가 기본 데이터 경로를 사용하는 기본 모드로 지정될 수 있다.

본 발명의 제2 측면에 따르면, 명령어에 추가된 재구성 모드 비트에 따라 데이터 경로를 재구성하는 재구성 제어 유닛이 구비된 재구성 가능 프로세서가 제공된다. 그리고, 상기 재구성 제어 유닛은, 응용 프로그램에 사용된 N 개의 명령어 중에서 선택된 M 개($M < N$)의 재구성 모드를 저장하며, 응용 프로그램 실행시에 상기 M 개의 재구성 모드를 식별하기 위해 각 명령어에 추가되는 재구성 모드 비트를 해석하여 당해 명령어에 대응하는 데이터 경로를 실시간으로 재구성하며, 상기 재구성 모드로 선택되지 아니한 명령어는 기본 데이터 경로를 사용한다.

이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예를 설명토록 한다.

도 3은 본 발명의 바람직한 실시예에 따른 재구성 가능한 프로세서의 재구성 방법을 순차적으로 도시한 것으로서, 재구성 제어 유닛에 의하여 수행될 수 있다.

본 발명의 바람직한 실시예에 따른 재구성 가능한 프로세서의 재구성 방법은 프로세서의 재구성을 응용 프로그램의 동작 시에 요구되는 실시간 재구성과 응용프로그램 수행 전의 선 재구성(Pre-Configuration)으로 구분되어 실행된다.

먼저, 선 재구성을 위한 단계로서, 응용 프로그램의 명령어 구조를 분석하여, 사용된 명령어의 빈도 수 및/또는 요구 사이클 등을 기반으로 전체 N가지의 재구성 가능 방법으로부터 M ($M < N$) 개의 최적 모드를 찾아낸다(S310).

프로세서의 프로그램은 각각의 프로세서의 명령어 조합으로 이루어지며, 이들 명령어는 응용제품군에 따라 그 명령어의 사용 빈도가 현저하게 달라질 뿐만 아니라, 요구되는 명령어도 달라진다. 이에, 응용 프로그램에 사용된 명령어의 조합을 분석함으로써 요구되는 데이터 경로를 최적화하여 M개의 재구성 모드를 생성할 수 있는 것이다.

설명의 편의를 도모하고자, 응용 프로그램 작성시에 표 1과 같이 7개의 명령어가 사용되었다고 가정한다.

[표 1]

| 명령어 | 사용빈도 | 기본 데이터 경로 사용시 사이클 수 | 총 사이클 수 | 재구성시 총 사이클 수 | 감소 사이클 수 |
|--------|------|---------------------|---------|--------------|----------|
| Inst 1 | 20 | 2 | 40 | 20 | 20 |
| Inst 2 | 15 | 2 | 30 | 15 | 15 |
| Inst 3 | 10 | 1 | 10 | 10 | 0 |
| Inst 4 | 10 | 2 | 20 | 10 | 10 |
| Inst 5 | 8 | 3 | 24 | 8 | 16 |
| Inst 6 | 5 | 2 | 10 | 5 | 5 |
| Inst 7 | 4 | 4 | 16 | 4 | 12 |

표 1을 참조하면, 명령어(Inst1)는 당해 응용 프로그램에 20회 사용되고, 기본 데이터 경로를 사용할 경우 2회의 단위 연산이 요구되어 2 사이클이 필요하다. 따라서, 명령어(Inst1)는 응용 프로그램 전체에서 총 40 사이클(20회 × 2 사이클)이 요구됨을 알 수 있다. 그런데, 명령어(Inst1)를 2 개의 단위 연산을 1 사이클에 수행하도록 전술한 도 2의 데이터 경로를 재구성할 경우에는, 응용 프로그램 전체에 걸쳐 20 사이클이 요구되어, 재구성 이전보다 20 사이클이 감소한다.

이러한 점에 착안하여, 전술한 단계(S310)에서 표 1에 연결된 7개(N=7)의 명령어 중에서 기본 경로를 이용하는 모드 외에 3개의 재구성 모드를 추가로 선택함으로써, 다음에 설명하는 바와 같이 기본 경로를 이용하는 기본 모드를 포함하여 총 4개(M=4)의 재구성 모드를 정의할 수 있다. 이 때, 각 재구성 모드를 구분하여 지시하기 위해 명령어에 필요한 추가 비트는 2 비트가 된다.

1. 명령어 사용 빈도 기준

위 7개의 명령어 중 사용 빈도가 높은 순으로 정리하면, Inst1(20)>Inst2(15)>Inst3,Inst4(10)>Inst5(8)>Inst6(5)>Inst7(4)이다. 여기서, 명령어(Inst3)는 기본 데이터 경로를 이용하여 한 사이클에 수행할 수 있는 명령어이므로 제외하고, Inst1, Inst2, Inst4를 위한 재구성 모드를 선택한다.

2. 명령어가 요구하는 사이클 수 기준

명령어 사이클 수에 따라 정리하면, Inst7(4)>Inst5(3)>Inst1,Inst2,Inst4, Inst6(2)>Inst3(1)이다. 따라서, 사이클 수를 많이 요구하는 Inst7, Inst5를 먼저 선택하고, 같은 수의 사이클을 요구하는 Inst1, Inst2, Inst4, Inst6 중에서는 빈도수가 높은 Inst1을 선택할 수 있다. 즉, 명령어가 요구하는 사이클 수 기준으로 할 경우에는, Inst7, Inst5, Inst1이 선택된다.

3. 사용 빈도 및 명령어 요구 사이클 기준

사용 빈도와 사이클 수를 같이 고려할 경우에는, 기본 데이터 경로를 사용했을 때 응용 프로그램을 모두 수행함에 있어서 각 명령어가 요구하는 총 사이클 수를 비교하여 선택한다. 이때, 사용 빈도수와 기본 데이터 경로 이용시 요구 사이클 수의 곱은 표 1의 총 사이클수에 해당하며, 표 1의 총 사이클수를 비교하면 Inst1(40)>Inst2(30)>Inst5(24)>Inst4(20)>Inst7(16)>Inst3,Inst6(10)이다. 따라서, 본 기준에 따르면 Inst1, Inst2, Inst5가 재구성 모드로 선택한다.

4. 재구성시 감소되는 사이클 수 기준

각 명령어를 기본 데이터 경로를 사용했을 때 요구되는 총 사이클 수에서 재구성시 요구되는 총 사이클 수를 차감하면, 기 설명한 바와 같이 재구성시 감소되는 총 사이클 수를 계산 할 수 있다. 이러한 감소 사이클 수에 따라 명령어를 정렬하면, Inst1(20)>Inst5(16)>Inst2(15)>Inst7(12)>Inst4(10)>Inst6(5)>Inst3(0)이다. 따라서, 본 기준에 따르면, Inst1, Inst5, Inst2를 재구성 모드로 선택함으로써, 재구성 시에 51개의 사이클을 줄일 수 있다.

전술한 단계(S310)에 후속하여, 재구성 제어 유닛이 상기 선택된 M 개의 모드를 식별하여 해당 데이터 경로를 구성할 수 있도록 재구성 제어 유닛의 재구성 모드를 설정한다(S320). 이러한 재구성 모드의 설정은 상기 재구성 모드에 관한 정보를 재구성 제어 유닛에 다운로드함으로써 수행될 수 있다. 이어서, log₂M개의 재구성 모드 비트를 각 명령어에 추가하여 (S330), 명령어를 수신한 재구성 제어 유닛이 재구성 모드를 식별할 수 있도록 한다.

마지막으로, 실시간 재구성 단계로서, 응용프로그램 수행 시에 재구성 제어 유닛이 재구성 모드 비트를 해석하여 데이터 경로의 재구성을 실시간으로 수행한다(S340). 이 때, 상기 재구성 모드로 선택되지 아니한 명령어는 기본 데이터 경로를 사용하는 기본 모드로 재구성 모드 비트가 설정되므로, 기본 데이터 경로를 사용한다.

이와 같이, 본 발명의 바람직한 실시예에 따라 생성된 M 개의 재구성 모드는 응용 프로그램의 수행 전에 재구성 제어 유닛에 설정되어 선 재구성되며, 각각의 수행 명령어는 $\log_2 M$ 개의 추가 비트만을 가지고 실시간 재구성을 가능하게 한다.

한편, 전술한 본 발명의 바람직한 실시예에서는 재구성 가능 프로세서의 재구성 방법에 관하여 설명하였으나, 본 발명은 이러한 방법의 각 단계 또는 기능을 구비하거나 그 결과가 프로그래밍(다운로드)된 재구성 가능 프로세서의 형태로 제공될 수 있다.

이상에서 본 발명에 따른 바람직한 실시예를 설명하였으나, 이는 예시적인 것에 불과하며 당해 분야에서 통상적 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 여타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 보호 범위는 이하의 특허청구범위에 의해서 정해져야 할 것이다.

발명의 효과

이상 설명한 바와 같이, 본 발명에 따르면 응용 프로그램에서 요구되는 명령어를 기반으로 프로세서의 재구성 모드를 먼저 설정하는 선 재구성을 수행함으로써, 실제 응용 프로그램이 수행될 때의 실시간 재구성시에 먼저 결정된 모드만을 이용하여 재구성에 요구되는 코드의 사이즈 및 하드웨어를 경량화시킬 수 있다. 즉, 데이터 경로의 재구성에 있어서 선재구성과 실시간 재구성의 두 단계 재구성 구조를 이용함으로써, 재구성의 유연성 증대와 함께 프로그램 코드 사이즈를 줄일 수 있는 것이다.

또한, 모드를 미리 설정함으로써 응용 프로그램의 특징에 맞게 재구성하여 사용할 수 있는 유연성을 부여하며, 소프트웨어를 이용한 선 재구성 방법을 이용함으로써 응용 제품군에 종속적으로 프로세서를 재구성하여 프로세서의 응용 영역을 확장할 수 있다.

도면의 간단한 설명

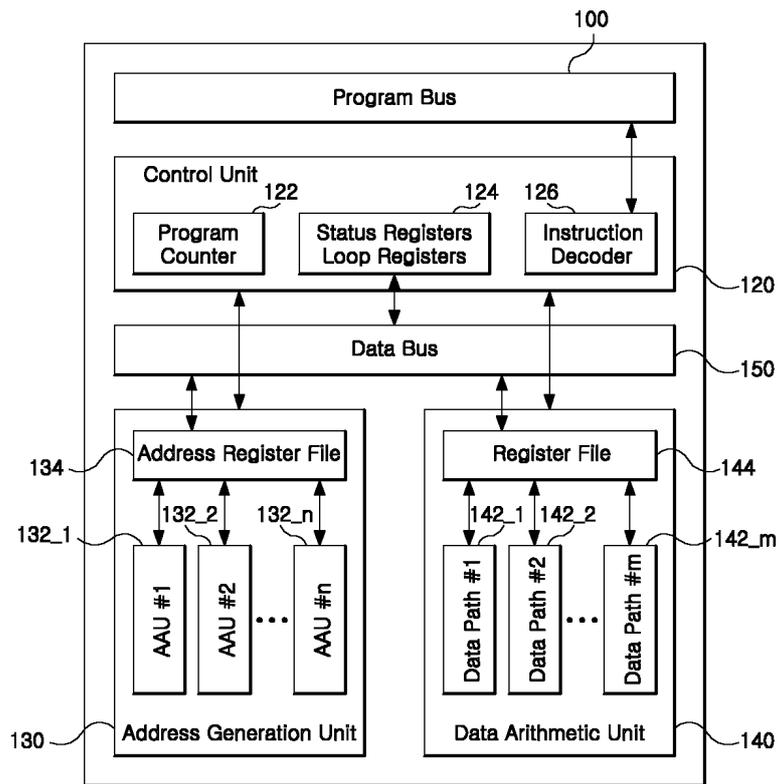
도 1은 통상의 병렬 처리 디지털 신호처리기(DSP)의 구성도.

도 2는 디지털 신호처리기에서 데이터 경로의 재구성이 가능한 데이터 경로 유닛의 구성도.

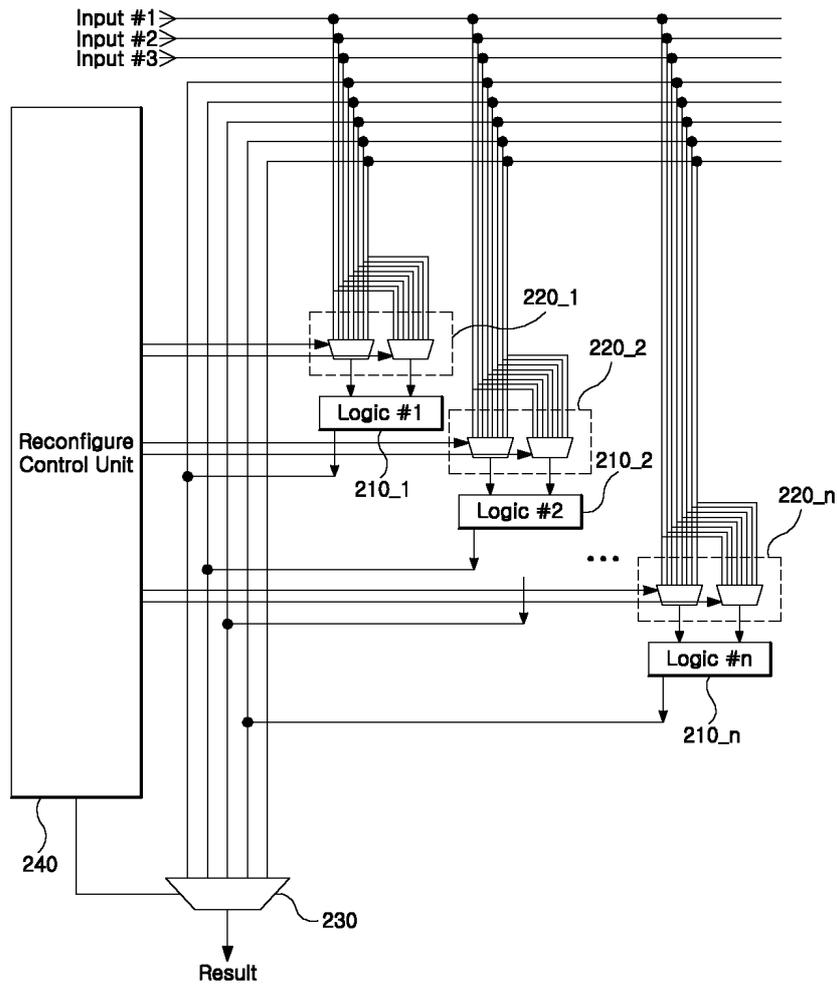
도 3은 본 발명의 바람직한 실시예에 따른 재구성 가능한 프로세서의 재구성 방법을 순차적으로 도시한 흐름도.

도면

도면1



도면2



도면3

