스트림-기반 프로그래밍으로 FPGA 기반

임베디드 애플리케이션 가속

임펄스 C는 잘 정의된 데이터 통신과 메시지 패싱, 동기화 매커니즘을 사용해 보다 쉽게 혼성 소프트웨어 • 하드웨어 알고리즘을 비롯한 고도의 병렬 알고리즘을 생성할 수 있다.

글 : 데이비드 펠러린(David Pellerin) CTO, Impulse Accelerated Technologies
자료제공 : 자일링스 www.xilinx.com

최근 기술 발전으로 인해 소프트웨어 애플리케이션 개발자는 소프트웨어-하드웨어 컴파일러를 통해 연산 집약적 알고리즘을 전용 하드웨어 가속기로 보다 쉽게 이동시킬 수 있다. 하지만 전통적인 프로세서 타깃 애플리케이션 구현에 익숙한 소프트웨어 프로그래머의 경우, FPGA를 타깃으로 할 때 병렬처리의 장점을 충분히 활용하지 못할 수도 있다.

이 글에서는 복잡한 혼성 하드웨어/소프트웨어 애플리케이션을 구현하는데 도움이 되는 소프트웨어-하드웨어 툴의 핵심 내용을 설명하고, FPGA 기반 하드웨어 가속 애플리케이션에 이상적인 C 언어 프로그래밍 모델을 기술하고자 한다.

C 언어 병렬 처리

C 프로그래밍 언어를 사용한 병렬 하드웨어 애플리케이션을 기술한다는 것은 어려운 일이다. 표준 ANSI C는 순차적 프로그래밍 언어로 상대적으로 단순하기 때문에 병렬식 알고리즘에 적합하지 않으며 어떤 면에서 이해하기 어려울 수 있기 때문이다. 따라서 확실한 병렬처리를 지원할 수 있는 언어를 확장할 수 없거나 다른 시퀀셜 C 스테이트먼트에서 로우-레벨 병렬 처리를 추출할 수 있는 특화된 컴파일러에 액세스 할 수 없다면, 병렬 시스템을 기술하거나 병렬 연산을 생성하는데 표준 C를 이용할 수 없다.

시스템이나 애플리케이션 레벨의 병렬 처리는 다양한 병렬 프로그래밍 모델을 지원하는 C로 확장할 수 있

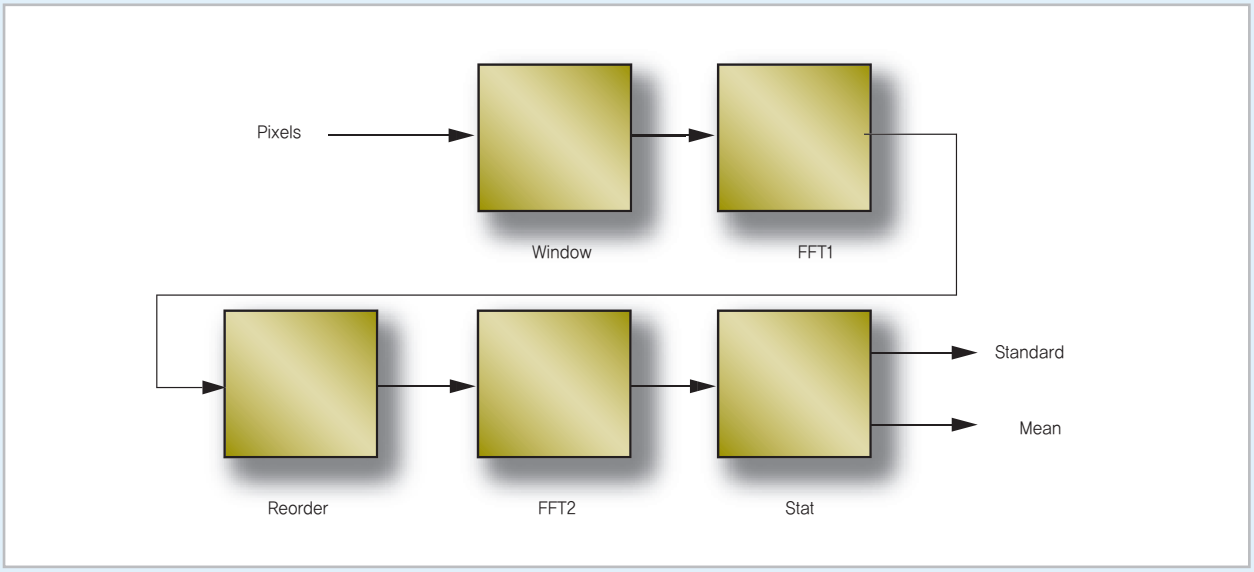

그림 1. 스트리밍 프로그래밍 모델을 통해 개발한 복잡한 이미지 필터 애플리케이션

다. 가장 일반적인 확장 방법은 쓰레딩(Threading)이다. 표준 C 언어는 쓰레드를 지원하지 않지만 수많은 소프트웨어 프로그래머들은 C-언어 쓰레드 라이브러리(POSIX 쓰레드, 마이크로소프트 윈도우 쓰레딩 라이브러리와 같은)를 통해 다중 병렬 연산을 처리하고 쓰레드 애플리케이션에서 요구되는 뮤텍스(Mutexes), 조건부 변수 및 다른 요소들을 기술한다.

병렬 처리를 위한 덜 보편적인 C 라이브러리로 MPI(Message Passing Interface)가 있다. 이 프로그래밍 라이브러리는 병렬 컴퓨팅 클러스터와 하이 엔드 수퍼 컴퓨터 분야에서 주로 사용된다. MPI를 통해 C 프로그래머는 통신 및 동기화 메시지 패싱 기법을 사용해 연결된 프로세스 콜렉션과 같은 연산 집약적 애플리케이션을 기술할 수 있다. MPI 프로그래밍 모델은 컴퓨팅 클러스터의 한 노드에서 동작하는 소프트웨어 프로세스 메시지를 통과할 수 있도록 지원하며, 다른 시스템 노드에서 동작하는 소프트웨어 프로세스와의 동작을 조율할 수 있다. MPI는 서로 다른 알고리즘과 연산이 각 노드에서 동작하거나, 동일한 소프트웨어 프로세스의 병렬 어레이가 서로 다른 데이터 세트에서 각각 동작하는 애플리케이션에서 의미를 가지고 있다.

스트리밍은 MPI와 유사한 특성을 가지고 있는 또 다른 프로그래밍 모델 중 하나로, 관련된 다중 프로세스 간 데이터 통신 및 동기화가 가능하다. 스트리밍은 레이턴시는 낮지만 복잡한 데이터 연산을 요구하는 HPEC(High-Performance Embedded Computing) 분야의 수많은 애플리케이션에 유용한 프로그래밍 모델로, 버텍스-4 FX 디바이스와 같은 혼성 프로세서/FPGA 플랫폼에 적합하다.

임펄스 C 스트리밍 프로그래밍 모델

임펄스 C는 병렬 프로그래밍 함수 라이브러리로 표준 ANSI C에 기반한 컴파일러다. 임펄스 C는 병렬 프로그래밍을 지원하도록 C 언어를 확장하는 것은 물론 디버거 및 프로파일러와 같은 표준 C 툴과의 상호 호환도 지원하다. 임펄스 C는 데이터 통신과 메시지 패싱, 동기화 매커니즘을 사용해 보다 쉽게 혼성 소프트웨어 · 하드웨어 알고리즘을 비롯한 고도의 병렬 알고리즘을 생성할 수 있다.

임펄스 C 스트리밍 애플리케이션에서 하드웨어 및 소프트웨어 프로세스는 주로 하드웨어에 직접 구현된 버퍼링 데이터 스트림을 통해 통신이 이루어진다. 이 버퍼링 데이터는 컴파일러

로 생성된 듀얼-클럭 FIFO 인터페이스를 통해 구현되며, 사이클-바이-사이클 동기화를 하지 않고 상대적으로 높은 추상화 레벨로 병렬 애플리케이션을 구현할 수 있다.

그림 1은 복잡한 이미지 분석을 위해 멀티플 프로세스 구현이 필요한 애플리케이션에의 분할 방법을 보여주고 있다. 이 개별 프로세스는 하드웨어 및 소프트웨어 요소들과의 통신이나 FPGA에서 구현될 수 있다. 병렬 및 파이프라인 프로세스 콜렉션은 이미지 프로세싱 동작을 수행하며, 각 프로세스는 하나 또는 그 이상의 C 서브루틴을 사용해 기술된다.

예를 들어 임베디드 PowerPC 405 프로세서 상의 애플리케이션 소프트웨어에서 임펄스 C 기능 호출은 데이터 스트림을 열거나 닫기 위해 혹은 스트림에서 데이터를 읽거나 쓰기 위해 사용돼 스트림 상태를 점검할 수 있으며, 프로세스-투-프로세스 메시지 전송이나 결과 계산을 위해 관련된 임펄스 C 함수 사용을 지원한다. 버텍스-4 FX 디바이스의 경우 APU(Auxiliary Peripheral Unit) 인터페이스를 이용해 읽기 및 쓰기 스트림을 규정할 수 있다. APU 인터페이스는 싱글 트랜잭션에서 128bit의 데이터 전송이 가능한 고성능 소프트웨어-하드웨어 시리얼 인터페이스를 제공한다.

임펄스 C는 데이터 플로우 중심 스트리밍 애플리케이션을 위해 디자인되었으나, 통신 매커니즘 공유 메모리를 사용하거나 프로그래밍 모델을 선택할 수 있도록 충분한 유연성을 제공한다. 임펄스 C 프로그래머는 애플리케이션 요구조건에 의존하거나 사용하고자 하는 프로그래밍 플랫폼의 아키텍처 조건에 따라 프로그래밍 모델을 선택하게 될 것이다. 버텍스-4 FX FPGA를 포함한 일부 플랫폼의 데이터 스트리밍 기법은 프로세서, FPGA 가속기 간 통신에 있어 높은 처리량을 제공한다. 다른 플랫폼에서의 데이터 전송은 공유 메모리 접근방법을 통해 가속하는 것이 보다 적합할 것이다.

임펄스 C 스트리밍 프로그래밍 모델의 핵심은 프로세스와 스트림이다. 프로세스는 독립적으로 동기화되며, 동시에 해당 애플리케이션 세그먼트를 실행한다. 표준 C 서브셋은 데이터를 받아들이고 연산 수행 및 출력을 생성함으로써 해당 애플리케이션 동작 및 하드웨어 쓰기 프로세스를 실행하기 위해 사용된다. 스트림은 프로세스 간 데이터 통신을 관리하기 위해 버퍼링 및 자가 동기화 기법을 제공한다.

앞에서 언급한 파이프라인 이미지 필터와 같은 전형적인 애플리케이션의 경우 해당 애플리케이션에서 프로세스 된 데이터는 스트림 수단(일부 경우에는 메시지나 공유 메모리)을 통해 프로세스에서 프로세스까지 흐르게 될 것이며, C 애플리케이션에서 생성된 FIFO의 폭과 깊이를 포함한 각 스트림의 특성을 정의할 수 있다. 실제 스트림 동작은 그림 2에 나타낸 것과 같이 C 함수를 통해 기술된다.

광범위한 애플리케이션

스트리밍 프로그래밍 모델을 사용해 제한 없이 수많은 애플리케이션을 구현할 수 있지만, 가장 효율적인 애플리케이션은 다음과 같은 특성을 일부 혹은 모두 가지고 있어야 한다.

- 데이터 소스와 프로세싱 요소 간 높은 데이터 전송속도. 데이터 소스는 임베디드 PowerPC와 같은 프로세서가 될 수도 있으며, 혹은 카메라나 다른 데이터 수집 디바이스와 같은 직접적인 하드웨어 인터페이스가 될 수도 있다.
- 메시지를 통해 요구되는 임시적 동기화와 주로 패스되는 데이터를 통해 통신하는 독립적인 멀티플 프로세스
- 스트림 데이터 패킷 크기는 전형적으로 8bit~128bit이다.
- 관련되어 있거나 독립된 멀티플 연산은 동일한 데이터 스트림 상에서 수행되어야 한다. 이 특성을 통해 알고리즘은 시스템 레벨 파이프라인을 사용해 병렬화되거나, 데이터를 분리시키고 연결된 병렬 프로세스 어레이를 생성함으로써 병렬화된다.
- 일반적으로 C 어레이 형태로 나타내는 로컬 및 공유 메모리는 데이터 스토리지 및 계수, 그리고 다른 상수 등을 위해 사용되거나 연산 결과를 누적시키기 위해 사용될 수 있다.

임펄스 C는 다음과 같은 다양한 분야의 애플리케이션을 처리하

```
co_stream_open(sample_stream, O_RDONLY, INT_TYPE(32));
co_stream_open(result_stream, O_RDONLY, INT_TYPE(32));
do {
        co_stream_read(sample_stream,&sr,sizeof(int32));

        ...//Process the data here

        co_stream_read(result_stream,&sr,sizeof(int32));
} while (...);
```

그림 2. 스트리밍 동작을 기술하기 위해 C 라이브러리 함수를 사용할 수 있다.

기 위해 특별히 설계되었다.

- 시그널 및 이미지 프로세싱
- 음성 및 데이터 통신
- 데이터 압축 및 암호화
- 과학적 연산
- 생물정보학
- 회계 모델링

병렬 FPGA 하드웨어 생성

하드웨어/소프트웨어 시스템이나 멀티 프로세스를 생성할 때 C로 쓰여진 애플리케이션 스트리밍 컴파일러는 2가지 주요 기능을 수행해야 한다. 첫째, 컴파일러는 각 병렬 프로세스(단일 C 서브루틴이나 서브루틴 콜렉션으로 각각 나타낼 수 있다) 분석이 가능해야 하며, 다른 시퀀셜 C 언어 스테이트먼트에서 로우 레벨 병렬 처리를 추출할 수 있어야 한다. 스트리밍 컴파일러는 하이 레벨에서 루프와 같은 구조를 평가할 수 있어야 하며, 언롤링, 파이프라이닝, 명령 스케쥴링과 같은 기법을 사용해 처리량을 증가시켜야 한다.

임펄스 C 컴파일러는 브랜치 구조, 할당, 중간 동작들에 대한 분석 및 실행 환경에 대한 최적화를 통해 시간 단축이 가능하다. 관련 임펄스 툴은 반복적인 최적화를 지원하는데 필요한 생성된 병렬 구조를 상호 분석할 수 있다.

스트리밍 컴파일러의 두 번째 주요 기능은 필수적인 프로세스-투-프로세스 인터페이스를 생성하는 것으로, 이 인터페이스는 FIFO 버퍼와 같이 직접적인 하드웨어 커넥션으로 구현되거나 혹은 표준 버스 및 플랫폼에 특화된 시리얼, DMA 커넥션처럼 하드웨어-투-프로세서 인터페이스로 구현할 수 있다.

임펄스 C 컴파일러는 시스템 생성에 필요한 주요 기능을 모두 지원하며, VHDL이나 Verilog 하드웨어 생성을 통해 자일링스 플랫폼 스튜디오(Xilinx Platform Studio)나 ISE 소프트웨어 툴을 이용한 FPGA 하드웨어 합성이 가능하다. 프로세서 측면에서 컴파일러는 임베디드 PowerPC나 MicroBlaze 프로세서, 혹은 보드-레벨 플랫폼에서 지원되는 외부 프로세서를 위해 사용 가능한 런타임 라이브러리를 생성한다.

결론

스트림 기반 프로그래밍은 혼성 소프트웨어 · 하드웨어 애플리케이션을 생성할 수 있도록 고성능 임베디드 시스템 개발자에게 대안적 방법을 제공한다. 임펄스 C를 통한 스트리밍 프로그래밍 모델은 FPGA 기반 플랫폼에 이상적이며, 특히 인접 FPGA 패브릭 내 임베디드 프로세서 및 전용 코프로세서 간 고대역폭 인터페이스를 제공하는 플랫폼에 더욱 이상적이다. 버텍스-4 FX FPGA는 스트리밍 프로그래밍 모델을 지원하는 플랫폼 중 하나이며, 복잡한 임베디드 소프트웨어 애플리케이션의 하드웨어 가속을 위한 맞춤형 솔루션이다.

스트림 중심 프로그래밍과 C에서의 FPGA 컴필레이션에 대한 상세 정보는 www.impulsec.com/xilinx에서 제공된다. RTime