(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0055672
(43) 공개일자 2023년04월26일

(51) 국제특허분류(Int. Cl.)
G06F 12/0815 (2016.01) *G06F 12/0811* (2016.01)
G06F 12/084 (2016.01) *G06F 12/0897* (2016.01)
(52) CPC특허분류
G06F 12/0815 (2013.01)
G06F 12/0811 (2013.01)
(21) 출원번호 10-2021-0139337
(22) 출원일자 2021년10월19일
심사청구일자 2021년10월19일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
신상원
서울특별시 송파구 올림픽로 435, 223동 202호
서태원
경기도 남양주시 순화궁로 18, 4110동 902호
(74) 대리인
김홍석

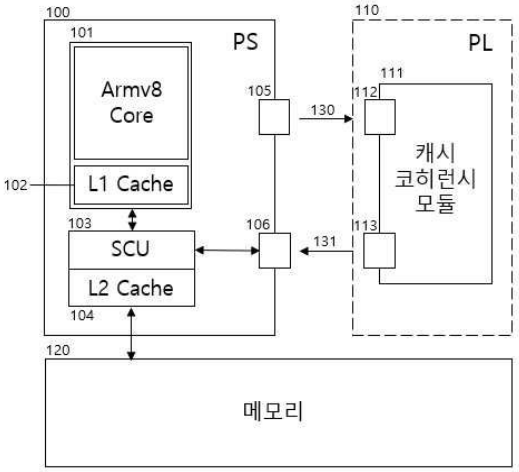
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **FPGA 기반 캐시 코히런시 방법 및 이를 수행하는 장치**

(57) 요 약

본 발명에 따른 FPGA 기반 캐시 코히런시 기능을 수행하는 장치는 인에이블 비트(Enable bit) 값이 1로 설정되면, 타겟 주소의 읽기(read)에 대한 PL AXI 신호를 전송하고, 상기 PL AXI 신호에 대한 응답을 수신하도록 구성된 캐시 코히런시 모듈; 및 상기 캐시 코히런시 모듈과 동작 가능하게 결합되고, 상기 PL AXI 신호가 수신되면 코히런시 유지 요청을 하는 함수를 통해 상기 타겟 주소와 모드 비트를 설정하고, 상기 인에이블 비트를 1로 설정하여 상기 캐시 코히런시 모듈을 동작시키는 프로세싱 시스템(Processing System: PS)을 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 12/084 (2013.01)
G06F 12/0897 (2013.01)
G06F 2212/1016 (2013.01)
G06F 2212/1052 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126288
과제번호	2019-0-00533-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보보호핵심원천기술개발(R&D,정보화)
연구과제명	컴퓨터 프로세서의 구조적 보안 취약점 검증 및 공격 탐지대응
기 여 율	1/1
과제수행기관명	고려대학교산학협력단
연구기간	2021.01.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

FPGA 기반 캐시 코히런시 기능을 수행하는 장치에 있어서,

인에이블 비트(Enable bit) 값이 1로 설정되면, 타겟 주소의 읽기(read)에 대한 PL AXI 신호를 전송하고, 상기 PL AXI 신호에 대한 응답을 수신하도록 구성된 캐시 코히런시 모듈; 및

상기 캐시 코히런시 모듈과 동작 가능하게 결합되고, 상기 PL AXI 신호가 수신되면 코히런시 유지 요청을 하는 함수를 통해 상기 타겟 주소와 모드 비트를 설정하고, 상기 인에이블 비트를 1로 설정하여 상기 캐시 코히런시 모듈을 동작시키는 프로세싱 시스템(Processing System: PS) 을 포함하는, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

청구항 2

제1 항에 있어서,

상기 캐시 코히런시 모듈은,

상기 응답이 상기 프로세싱 시스템으로부터 수신되면 타겟 주소의 쓰기(write)를 위한 제2 PL AXI 신호를 상기 프로세싱 시스템으로 전송하고,

상기 타겟 주소는 캐시 코히런시를 유지하기 위한 주소를 담아 저장하도록 구성되는, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

청구항 3

제1 항에 있어서,

상기 프로세싱 시스템은,

상기 캐시 코히런시 모듈의 동작이 정상적으로 완료되면 상기 인에이블 비트를 0으로 설정하는, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

청구항 4

제1 항에 있어서,

상기 캐시 코히런시 모듈은,

상기 모드 비트에 기반하여 연속되는 읽기 및 쓰기 동작에 대한 AxPROT 값을 결정하며,

상기 모드 비트가 1로 설정된 경우 Non-Secure 모드로 읽기 후 Secure 모드로 쓰기 동작을 수행하는, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

청구항 5

제4 항에 있어서,

상기 캐시 코히런시 모듈은,

상기 모드 비트가 0으로 설정된 경우 Secure 모드로 읽기 후 Non-Secure 모드로 쓰기 동작을 수행하는, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

청구항 6

제1 항에 있어서,

상기 프로세싱 시스템은,

빠른 데이터 처리를 위한 고유의 L1 cache를 가지고 있는 적어도 하나의 코어들;

상기 코어들과 동작 가능하게 결합되고, 상기 코어들이 공유하도록 구성된 L2 cache; 및

상기 코어 및 상기 L2 cache와 동작 가능하게 결합되고, 상기 코어들의 각 L1 cache와 상기 코어들이 공유하는 상기 L2 cache 간의 데이터 일관성을 유지하도록 구성된 SCU를 포함하는, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

청구항 7

제6 항에 있어서,

상기 L2 cache와 동작 가능하게 결합되고, 정보를 저장하도록 구성된 메모리를 더 포함하고,

상기 L2 cache는 독점적으로 사용되는 상기 L1 cache와 달리 상기 코어들이 공유하고, 상기 L2 cache는 상기 L1 cache보다 데이터를 읽고 쓰는 속도가 느리고, 상기 메모리보다 데이터를 읽고 쓰는 속도가 빠르게 설정되는, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

청구항 8

제6 항에 있어서,

상기 캐시 코히런시 모듈은 프로그래머블 로직(PL)에 포함되고,

상기 프로세싱 시스템은,

상기 PL의 AXI Slave Port로 AMBA AXI4 프로토콜에 따라 PS AXI 신호를 전송하도록 구성된 AXI Master Port; 및

상기 PL의 AXI Master Port로부터 상기 PL AXI 신호를 수신하도록 구성된 ACP를 더 포함하는, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

청구항 9

제8 항에 있어서,

상기 ACP는 상기 AMBA AXI4 프로토콜을 따르며 상기 SCU에 연결되어, 상기 캐시 코히런시 모듈이 상기 L1 cache 및 상기 L2 cache의 데이터 일관성에 관하도록 구성된, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

청구항 10

제8 항에 있어서,

상기 프로그래머블 로직(PL)은,

상기 캐시 코히런시 모듈;

상기 캐시 코히런시 모듈과 동작 가능하게 결합되고, 상기 프로세싱 시스템의 상기 AXI Master Port로부터 PS AXI 신호를 수신하도록 구성된 AXI Slave Port; 및

상기 캐시 코히런시 모듈과 동작 가능하게 결합되고, 상기 프로세싱 시스템의 상기 ACP로 상기 PL AXI 신호를 전송하도록 구성된 상기 AXI Master Port를 포함하는, FPGA 기반 캐시 코히런시 기능을 수행하는 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 컴퓨터 보안에 관한 것이며, 보다 상세하게는 ARM processor 기반의 FPGA (field programmable gate array) 기반의 하드웨어 로직을 통해 Cache에 보안 설정값이 다른 읽기/쓰기를 진행하여 data coherency를 보장하는 방법에 관한 것이다.

배 경 기 술

[0002] Arm 아키텍처 기반 프로세서는 오늘날 많은 임베디드 시스템에서 사용 중이다. 최근 발표된 Arm 기반 프로세서는 성능 향상을 위해 여러 개의 코어 (멀티 코어)로 이루어져 있으며 각 코어는 독립적으로 프로그램을 수행한다.

[0003] 프로세서는 DRAM (메모리)에서 각 코어로 데이터를 읽고/쓰는 속도를 향상시키기 위해 CPU 내부에 빠르고 작은 메모리인 캐시를 가지고 있다. 특히 Arm 아키텍처 기반 멀티 코어는 각 코어가 독점적으로 사용하는 L1 Data, Instruction 캐시, 여러 코어가 공유하는 L2 캐시를 가진다. 이러한 구조에서 각 코어는 독립적으로 명령어를 수행하여, 메모리에 읽기/쓰기 요청을 개별적으로 보내기 때문에, 캐시는 이러한 요청들에 대해 데이터 일관성 (Data consistency)을 유지해야 한다.

[0004] Arm 아키텍처의 경우 데이터 일관성 유지를 위해 SCU(Snoop Control Unit) 하드웨어를 가지고 있으며, SCU는 각 코어의 메모리 요청을 스누핑 하여 L1, L2, 메모리 간의 일관성을 유지한다. 또한, 코어뿐만 아니라 외부 I/O 장치들에서도 메모리 읽기/쓰기 요청이 가능하며, 이 요청을 ACP(Accelerator Coherency Port)를 통해 SCU로 전달된다.

[0005] Arm TrustZone 기술은 하드웨어를 논리적으로 Secure world와 Normal world로 분리하는 기술이다. Secure world는 모든 하드웨어 자원을 포함하며, Normal world는 일부 하드웨어 자원을 포함한다. 코어는 SCR(Secure Configuration Register)의 NS bit 설정값이 0이면 Secure 모드로 동작하고 Secure world에 접근이 가능하며, NS bit 설정값이 1이면 Non-Secure 모드로 동작하고 Normal world에 접근이 가능하다. 메모리는 Secure 영역과 Non-Secure 영역으로 구분되며, Secure 영역은 Secure world에만 존재하여 코어는 Secure 모드로만 접근이 가능하며, Non-Secure 영역은 Secure world와 Normal world에 존재하여 코어는 각 world에 맞는 모드로 접근이 가능하다.

[0006] Non-Secure 메모리 영역의 동일한 주소에 대해서 코어가 Secure 모드로 접근할때와 Non-Secure 모드로 접근할 때, 캐시는 두 접근을 다른 주소에 대한 접근으로 인식하여 data coherency를 보장하지 않는다. 읽기 동작만 수행될 때에는 문제가 없지만, 쓰기 동작이 발생하면 같은 주소에 대해서 다른 데이터로 인식하는 문제가 발생할 수 있다는 문제점이 있다.

발명의 내용

해결하려는 과제

[0007] 본 발명은 상기와 같은 문제를 해결하기 위해 제안된 것으로, ARM processor 기반의 FPGA 기반의 하드웨어 로직을 통해 Cache에 보안 설정 값이 다른 읽기/쓰기를 진행하여 data coherency를 보장하고자 한다.

[0008] 또한, 본 발명이 해결하고자 하는 과제는 FPGA 기반으로 설계한 하드웨어로 ACP를 통해 다른 모드로 읽기 및 쓰기 동작을 수행하여 data coherency 문제를 해결하는 것이다.

[0009] 또한,본 발명이 해결하고자 하는 과제는 FPGA를 통해 Secure world와 Non-Secure world간의 data coherency 유지 과정에서 발생하는 메모리 트래픽과 소요시간을 줄이는 방법을 제공하는 것이다.

과제의 해결 수단

[0010] 본 발명에 따른 FPGA 기반 캐시 코히런시 기능을 수행하는 장치는 인에이블 비트(Enable bit) 값이 1로 설정되면, 타겟 주소의 읽기(read)에 대한 PL AXI 신호를 전송하고, 상기 PL AXI 신호에 대한 응답을 수신하도록 구성된 캐시 코히런시 모듈; 및 상기 캐시 코히런시 모듈과 동작 가능하게 결합되고, 상기 PL AXI 신호가 수신되면 코히런시 유지 요청을 하는 함수를 통해 상기 타겟 주소와 모드 비트를 설정하고, 상기 인에이블 비트를 1로 설정하여 상기 캐시 코히런시 모듈을 동작시키는 프로세싱 시스템(Processing System: PS)을 포함한다.

[0011] 실시 예에 따르면, 상기 캐시 코히런시 모듈은 상기 응답이 상기 프로세싱 시스템으로부터 수신되면 타겟 주소의 쓰기(write)를 위한 제2 PL AXI 신호를 상기 프로세싱 시스템으로 전송하고, 상기 타겟 주소는 캐시 코히런시를 유지하기 위한 주소를 담아 저장하도록 구성될 수 있다.

[0012] 실시 예에 따르면, 상기 프로세싱 시스템은 상기 캐시 코히런시 모듈의 동작이 정상적으로 완료되면 상기 인에이블 비트를 0으로 설정할 수 있다.

[0013] 실시 예에 따르면, 상기 캐시 코히런시 모듈은 상기 모드 비트에 기반하여 연속되는 읽기 및 쓰기 동작에 대한 AxPROT 값을 결정하며, 상기 모드 비트가 1로 설정된 경우 Non-Secure 모드로 읽기 후 Secure 모드로 쓰기 동

작을 수행할 수있다.

[0014] 실시 예에 따르면, 상기 캐시 코히런시 모듈은 상기 모드 비트가 0으로 설정된 경우 Secure 모드로 읽기 후 Non-Secure 모드로 쓰기 동작을 수행할 수 있다.

[0015] 실시 예에 따르면, 상기 프로세싱 시스템은 빠른 데이터 처리를 위한 고유의 L1 cache를 가지고 있는 적어도 하나의 코어들; 상기 코어들과 동작 가능하게 결합되고, 상기 코어들이 공유하도록 구성된 L2 cache; 및 상기 코어 및 상기 L2 cache와 동작 가능하게 결합되고, 상기 코어들의 각 L1 cache와 상기 코어들이 공유하는 상기 L2 cache 간의 데이터 일관성을 유지하도록 구성된 SCU를 포함할 수 있다.

[0016] 실시 예에 따르면, 상기 장치는 상기 L2 cache와 동작 가능하게 결합되고, 정보를 저장하도록 구성된 메모리를 더 포함하고, 상기 L2 cache는 독점적으로 사용되는 상기 L1 cache와 달리 상기 코어들이 공유하고, 상기 L2 cache는 상기 L1 cache보다 데이터를 읽고 쓰는 속도가 느리고, 상기 메모리보다 데이터를 읽고 쓰는 속도가 빠르게 설정될 수 있다.

[0017] 실시 예에 따르면, 상기 캐시 코히런시 모듈은 프로그래머블 로직(PL)에 포함되고, 상기 프로세싱 시스템은 상기 PL의 AXI Slave Port로 AMBA AXI4 프로토콜에 따라 PS AXI 신호를 전송하도록 구성된 AXI Master Port; 및 상기 PL의 AXI Master Port로부터 상기 PL AXI 신호를 수신하도록 구성된 ACP를 더 포함할 수 있다.

[0018] 실시 예에 따르면, 상기 ACP는 상기 AMBA AXI4 프로토콜을 따르며 상기 SCU에 연결되어, 상기 캐시 코히런시 모듈이 상기 L1 cache 및 상기 L2 cache의 데이터 일관성에 관하도록 구성될 수 있다.

[0019] 실시 예에 따르면, 상기 프로그래머블 로직(PL)은 상기 캐시 코히런시 모듈; 상기 캐시 코히런시 모듈과 동작 가능하게 결합되고, 상기 프로세싱 시스템의 상기 AXI Master Port로부터 PS AXI 신호를 수신하도록 구성된 AXI Slave Port; 및 상기 캐시 코히런시 모듈과 동작 가능하게 결합되고, 상기 프로세싱 시스템의 상기 ACP로 상기 PL AXI 신호를 전송하도록 구성된 상기 AXI Master Port를 포함할 수 있다.

발명의 효과

[0020] 본 발명은 AXI ACP와 AMBA 통신규약을 이용하여 Armv8 CPU에서 경량화된 data coherency 유지 역할을 제공할 수 있다.

[0021] 본 발명을 통하여 정보보호를 보장하면서 빠른 동작 수행을 해야하는 프로그램을 효과적으로 설계할 수 있는 환경을 제공한다.

[0022] 본 발명의 IP를 이용하여, 사용자는 함수처럼 간편하게 data coherency를 유지할 수 있다.

[0023] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다.

도면의 간단한 설명

[0024] 도 1은 본 발명의 제안하는 캐시 코히런시 모듈(111)을 포함한 시스템 전체 구조를 나타낸다.

도 2a는 본 발명에 따른 Non-Secure 모드와 Secure 모드에서 CPU의 읽기 및 쓰기 동작에 따른 코히런스 이슈와 관련된 개념도를 나타낸다.

도 2b는 본 발명에 따른 flush 방식 및 ACP 방식을 통해 캐시 코히런시 문제를 해결하는 방법의 개념도를 나타낸다.

도 3은 본 발명에 따른 AXI4 프로토콜의 채널을 나타낸다.

도 4는 본 발명에 따른 PL AXI 신호(131)와 관련된 신호 흐름도이다.

도 4는 본 발명에 따른 캐시 코히런시 모듈(111)이 수행하는 방법의 흐름도를 나타낸다.

도 6은 본 발명에 따른 캐시 코히런시 모듈(111)에 Target address의 코히런시 유지 요청을 하는 함수에 대한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0025] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다.

[0026] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는바, 특정 실시 예들을 도면에 예시하고 상세한 설명에 구체적으로 설명하고자 한다. 그러나 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0027] 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용한다.

[0028] 제1, 제2등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다.

[0029] 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0030] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미가 있다.

[0031] 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않아야 한다.

[0032] 이하의 설명에서 사용되는 구성요소에 대한 접미사 모듈, 블록 및 부는 명세서 작성의 용이함만이 고려되어 부여되거나 혼용되는 것으로서, 그 자체로 서로 구별되는 의미 또는 역할을 갖는 것은 아니다.

[0033] 이하, 본 발명의 바람직한 실시 예를 첨부한 도면을 참조하여 당해 분야에 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 설명한다. 하기에서 본 발명의 실시 예를 설명함에 있어, 관련된 공지의 기능 또는 공지의 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다.

[0034] 이하에서는, 본 발명에 따른 FPGA 기반 캐시 코히런시 방법 및 이를 수행하는 장치에 대해 도면을 참조하여 설명한다.

[0035] 도 1은 본 발명의 제안하는 캐시 코히런시 모듈(111)을 포함한 시스템 전체 구조를 나타낸다. 이와 관련하여, Arm 아키텍처 기반 프로세서는 오늘날 많은 임베디드 시스템에서 사용 중이다. 최근 발표된 Arm 기반 프로세서는 성능 향상을 위해 여러 개의 코어 (멀티 코어)로 이루어져 있으며 각 코어는 독립적으로 프로그램을 수행한다.

[0036] 프로세서는 DRAM (메모리)에서 각 코어로 데이터를 읽고/쓰는 속도를 향상시키기 위해 CPU 내부에 빠르고 작은 메모리인 캐시를 가지고 있다. 특히 Arm 아키텍처 기반 멀티 코어는 각 코어가 독점적으로 사용하는 L1 Data, Instruction 캐시, 여러 코어가 공유하는 L2 캐시를 가진다. 이러한 구조에서 각 코어는 독립적으로 명령어를 수행하여, 메모리에 읽기/쓰기 요청을 개별적으로 보내기 때문에, 캐시는 이러한 요청들에 대해 데이터 일관성 (Data consistency)을 유지해야 한다.

[0037] Arm 아키텍처의 경우 데이터 일관성 유지를 위해 SCU(Snoop Control Unit) 하드웨어를 가지고 있으며, SCU는 각 코어의 메모리 요청을 스누핑 하여 L1, L2, 메모리 간의 일관성을 유지한다. 또한, 코어뿐만 아니라 외부 I/O 장치들에서도 메모리 읽기/쓰기 요청이 가능하며, 이 요청을 ACP(Accelerator Coherency Port)를 통해 SCU로 전달된다.

[0038] Arm TrustZone 기술은 하드웨어를 논리적으로 Secure world와 Normal world로 분리하는 기술이다. Secure world는 모든 하드웨어 자원을 포함하며, Normal world는 일부 하드웨어 자원을 포함한다. 코어는 SCR(Secure Configuration Register)의 NS bit 설정값이 0이면 Secure 모드로 동작하고 Secure world에 접근이 가능하며, NS bit 설정값이 1이면 Non-Secure 모드로 동작하고 Normal world에 접근이 가능하다. 메모리는 Secure 영역과 Non-Secure 영역으로 구분되며, Secure 영역은 Secure world에만 존재하여 코어는 Secure 모드로만 접근이 가

능하며, Non-Secure 영역은 Secure world와 Normal world에 존재하여 코어는 각 world에 맞는 모드로 접근이 가능하다.

[0039] Non-Secure 메모리 영역의 동일한 주소에 대해서 코어가 Secure 모드로 접근할때와 Non-Secure 모드로 접근할 때, 캐시는 두 접근을 다른 주소에 대한 접근으로 인식하여 data coherency를 보장하지 않는다. 이와 관련하여, 도 2a는 본 발명에 따른 Non-Secure 모드와 Secure 모드에서 CPU의 읽기 및 쓰기 동작에 따른 코히런스 이슈와 관련된 개념도를 나타낸다. 도 2a를 참조하면, Non-Secure 모드 또는 Secure 모드인지에 따라 캐시에 다른 데이터가 저장될 수 있다. 이와 관련하여, 읽기 동작만 수행될 때에는 문제가 없지만, 쓰기 동작이 발생하면 같은 주소에 대해서 다른 데이터로 인식하는 문제가 발생할 수 있다. 따라서 쓰기 동작을 한 후에 명령어를 통해 캐시를 flush하여 데이터를 메모리로 이동시켜 coherency 문제를 해결한다.

[0040] 캐시를 flush하는 과정은 캐시의 데이터를 메모리로 이동시키는 것을 의미하며, 상황에 따라 coherency 문제가 발생할 가능성이 있는 데이터를 구분하지 않고 전체를 flush할 수도 있다. 또는, coherency 문제가 발생할 가능성이 있는 데이터에 대해서만 flush할 수도 있다. Flush가 포함된 두 가지 경우 모두 메모리 트래픽을 발생시키면서 추가적인 시간이 소요된다.

[0041] 프로세서 내부에는 다양한 장치 및 I/O가 있으며 버스를 통해 장치 간 통신을 한다. 각 프로세서 제조사마다 서로 다른 규약을 정의하여 사용하며, 대표적으로 Arm은 AMBA(Advanced Microcontroller Bus Architecture) 프로토콜을 사용한다. 2010년에 출시한 AMBA 4는 크게 ACE, AXI4, APB4 프로토콜을 지원하며, 본 발명에서는 AXI4 (Advanced eXtensible Interface 4) 프로토콜의 일부 신호를 조정하여 ACP로 Secure 모드 및 Non-Secure 모드로 읽기 및 쓰기 동작이 가능한 하드웨어를 설계할 수 있다.

[0042] 본 발명에서는 FPGA 기반으로 설계한 하드웨어로 ACP (Accelerator Coherency Port)를 통해 다른 모드로 읽기 및 쓰기 동작을 수행하여 data coherency 문제를 해결하는 기술을 제안한다. 이와 관련하여, 도 2b는 본 발명에 따른 flush 방식 및 ACP 방식을 통해 캐시 코히런시 문제를 해결하는 방법의 개념도를 나타낸다. 도 2b를 참조하면, 캐시의 데이터를 메모리로 이동시키는 flush 방식이 이루어질 수 있다. 또한, ACP (Accelerator Coherency Port)를 통해 다른 모드로 읽기 및 쓰기 동작을 수행하여 data coherency 문제를 해결할 수 있다. 이를 통해 데이터를 메모리로 이동시키는 과정에서 발생하는 메모리 트래픽과 소요되는 시간을 줄일 수 있다.

[0043] 한편, 이러한 캐시 코히런시 문제가 아주 자주 발생하는 상황에 해당하지 않는다. 따라서, flush 방식으로 캐시 코히런시 문제를 해결할 수 있다. 한편, ACP 방식은 AI 프로그램의 성능 향상에 주로 사용될 수 있다. 따라서, 본 발명에서는 flush 방식으로 캐시 코히런시 문제를 해결하되, AI 프로그램의 성능 향상을 위해서도 ACP 방식을 이용하는 방식을 제안하고자 한다. 이와 관련하여, 본 발명은 캐시 코히런시 수행 방법은 flush 방식에 비해 추가적으로 설계한 IP를 활용한다는 점에서 차별점이 있다.

[0044] 도 1을 참조하면, 본 발명에서 제안하는 하드웨어 IP(Intellectual Property)는 캐시 코히런시 모듈(111)이다. 메모리 복사 모듈(111)은 사용자가 설정한 값을 바탕으로 cache coherency 유지 동작을 수행한다. 사용자는 memory copy를 하기 위해 Enable bit, Target Address, Mode bit를 설정해야한다. Enable bit는 캐시 코히런시 모듈(111)의 작동 여부를 결정하며, 1로 설정할 경우 모듈이 동작하며 PL AXI 신호(131)가 발생한다. Target Address는 cache coherency를 유지하기 위한 주소를 담는다. Mode bit는 연속되는 읽기 및 쓰기 동작에 대한 AxPROT[1] 값을 결정하며, 1로 설정할 경우 Non-Secure 모드로 읽기 후 Secure 모드로 쓰기 동작을 수행하며 0으로 설정할 경우 Secure 모드로 읽기 후 Non-Secure 모드로 쓰기 동작을 수행한다. 각 동작에 대한 요청은 PL AXI 신호(131)를 통해 이루어진다.

[0045] 전술한 바와 같이, 도 1은 본 발명의 제안하는 캐시 코히런시 모듈(111)을 포함한 시스템 전체 구조를 나타낸다. 한편, 도 3은 본 발명에 따른 AXI4 프로토콜의 채널을 나타낸다. 도 4는 본 발명에 따른 PL AXI 신호(131)와 관련된 신호 흐름도이다. 또한, 도 5는 본 발명에 따른 캐시 코히런시 모듈(111)이 수행하는 방법의 흐름도를 나타낸다.

[0046] 도 1을 참조하면 cache coherency를 유지하는 장치는 크게 프로세싱 시스템(Processing System, PS)(100), 프로그래머블 로직(Programmable Logic, PL)(110) 및 메모리(120)로 이루어져 있다.

[0047] PS(100)는 Armv8 core(101), L1 cache(102), SCU(103), L2 cache(104), AXI Master Port(105), ACP(106)를 포함한다. 여기서, AXI 및 ACP(106)는 각각 dvanced eXtensible Interface 및 Accelerator Coherency Port를 의미한다. PL(110)은 캐시 코히런시 모듈(111), AXI Slave Port(112), AXI Master Port(113)를 포함한다. PS(100)의 AXI Master Port(105)는 PL(110)의 AXI Slave Port(112)로 PS AXI 신호(130)를, PL(110)의 AXI

Master Port(113)는 PS(100)의 ACP(106)로 PL AXI 신호(131)를 보낸다.

[0048] PS(100)는 Processing System의 약자로 프로세서를 나타내며, Armv8 core(101), L1 cache(102), SCU(103), L2 cache(104), AXI Master Port(105), ACP(106)를 포함한다. 각 Armv8 core(101)는 빠른 데이터 처리를 위한 고유의 L1 cache(102)를 가지고 있다. SCU(103)는 각 Armv8 core(101)들의 각 L1 cache(102) 와 코어들이 공유하는 L2 cache(104) 간의 데이터 일관성을 유지하는 역할을 한다. 데이터 일관성 유지란 여러 Armv8 core(101)에서 데이터 쓰기가 수행될 때 공유하고 있는 데이터의 불일치를 해결하는 일련의 과정을 말한다. L2 cache(104)는 독점적으로 사용되는 L1 cache(102)와 달리 Armv8 core(101)들이 공유한다. L2 cache(104)는 L1 cache(102)보다 데이터를 읽고 쓰는데 느리지만, 메모리(105)보다 빠르다. AXI Master Port(105)는 사용자가 캐시 코히런시 모듈(111)에 데이터를 보내기 위해 사용하는 인터페이스로 AMBA AXI4 프로토콜에 따라 PS AXI 신호(130)를 보낸다. ACP(106)는 AMBA AXI4 프로토콜을 따르며 SCU(103)에 연결되어 있기 때문에, 캐시 코히런시 모듈(111)은 L1 cache(102) 및 L2 cache(104)의 데이터 일관성에 관여할 수 있다.

[0049] PL(110)은 Programmable Logic의 약자이며, 본 발명에서 제안한 캐시 코히런시 모듈(111), AXI Slave Port(112), AXI Master Port(113)를 포함한다.

[0050] 캐시 코히런시 모듈(111)을 통해 사용자는 다른 보안 모드에 따른 캐시 코히런시를 메모리 접근 없이 유지할 수 있다.

[0051] 도 3은 전술한 바와 같이 AMBA AXI4 프로토콜에 대한 구조도이다.

[0052] 도 3의 AXI Master(200)은 AXI4 프로토콜에 따라 신호를 보내는 장치이고 AXI Slave(201)는 받는 장치이다. AXI 프로토콜은 크게 5개의 채널로 이루어져 있다. AR Channel(210)은 Read Address Channel로 AXI Master(200)에서 읽고자 하는 데이터의 주소에 관련된 채널이다. R Channel(211)은 Read Data Channel로 AXI Master(200)에서 읽고 싶은 데이터를 받는 채널이다. 이 두 채널(210, 211)은 읽기 작업과 관련되어 있다. AXI Master(200)가 AR Channel(210)로 신호를 보내면 AXI Slave(201)가 AR Channel(210)에서 받은 주소에 있는 데이터를 R Channel(211)로 보내준다. AW Channel(212)은 Write Address Channel로 AXI Master(200)에서 쓰고자 하는 데이터의 주소에 관련된 채널이다. W Channel(213)은 Write Data Channel로 AXI Master(200)에서 쓰고 싶은 데이터를 보내는 채널이다. B Channel(214)는 Write Response Channel로 AXI Slave(201)에서 받은 쓰기 신호에 응답하는 채널이다. 이 세 채널(212, 213, 214)은 쓰기 작업과 관련되어 있다. AXI Master(200)가 AW Channel(212)과 W Channel(213)로 신호를 보내면 AXI Slave(201)가 B Channel(214)로 쓰기 작업에 대한 상태를 보내준다.

[0053] 전술한 바와 같이, 도 4는 AXI 프로토콜의 채널에 대한 구조도이다.

[0054] 도 4의 AXI Sender(300)는 주소나 데이터를 전송하는 장치이고 AXI Receiver(301)는 주소나 데이터를 전송받는 장치이다. 이후 서술하는 “활성화”는 연결된 라인에 1을 보내는 것을 의미하며, “비활성화”는 0을 보내는 것을 의미한다. AXI Sender(300)가 전송할 Information(312)과 Configure(313)를 활성화한 상황에서 Valid(310)를 활성화한다. Ready(311)는 AXI Receiver(301)가 주소나 데이터를 전송받을 준비가 되면 활성화한다. Information(312)은 AXI Sender(300)가 전송할 주소나 데이터에 맞게 활성화한다. Configure(313)는 AXI Sender(300)가 전송에 대한 설정값에 맞게 활성화한다. Valid(310)와 Ready(311)가 동시에 활성화되면, AXI Receiver(301)은 Information(312)과 Configure(313)에 활성화된 정보를 인식하며 정보의 전달이 이루어진다. Valid(310)와 Ready(311)가 동시에 활성화되어 정보의 전달이 발생한 후, AXI Sender(300)는 Valid(310)를 비활성화하며 AXI Receiver(301)은 Ready(311)를 비활성화한다.

[0055] AR Channel(210)에서는 읽고자 하는 데이터의 주소인 ARADDR에 맞게 Information(312) 라인을 활성화하고 AW Channel(212)에서는 쓰고자 하는 데이터의 주소인 AWADDR에 맞게 Information(312) 라인을 활성화한다.

[0056] Configure(313)의 설정값은 Secure 모드 여부를 포함한다. 읽기 동작 시 AR Channel(210)의 설정값 중 ARPROT[1]을 0으로 설정하면 Secure 모드로 읽기 동작이 수행되고, 1로 설정하면 Non-Secure 모드로 읽기 동작이 수행된다. 쓰기 동작 시 AW Channel(212)의 설정값 중 AWPROT[1]을 0으로 설정하면 Secure 모드로 쓰기 동작을 수행하고, 1로 설정하면 Non-Secure 모드로 쓰기 동작을 수행한다. 캐시 코히런시 모듈(111)은 Mode bit에 따라 1일 경우 ARPROT[1]은 1로 설정하고 AWPROT[1]은 0으로 설정하며, 0일 경우 ARPROT[1]은 0으로 설정하고 AWPROT[1]은 1로 설정한다. AxPROT은 ARPROT와 AWPROT를 동시에 지칭할 때 사용한다.

[0057] 전술한 바와 같이, 도 5는 캐시 코히런시 모듈(111)의 동작 과정에 대한 순서도이다.

[0058] 캐시 코히런시 모듈(111)은 먼저 Enable bit의 값을 읽는다(S410). Enable bit가 1인지 확인한다(S411). Enable bit가 0이면 S410 단계로 돌아가 다시 Enable bit의 값을 읽고(S410), Enable bit가 1이면 PL AXI 신호(131)를 설정한다(S412). PL AXI 신호(131) 중 ARADDR은 Target address로, ARPROT는 Mode bit에 따라 설정한다. 설정이 완료되면 Target address 읽기에 대한 PL AXI 신호(131)를 전송한다(S413). Read에 대한 응답(read response)이 수신되지 않으면 S412 단계로 돌아가 다시 신호를 보내며, 수신이 확인되면 PL AXI 신호(131)를 설정한다(S415). PL AXI 신호(131) 중 AWADDR은 Target address로, AWPROT는 Mode bit에 따라 설정한다. 설정이 완료되면 Target address 쓰기에 대한 PL AXI 신호(131)를 전송한다(S416). Write에 대한 응답(write response)이 수신되지 않으면 S415 단계로 돌아가 다시 신호를 보내며, 수신이 확인되면 Enable bit를 0으로 설정하고(S418) 정상적으로 종료된다.

[0059] 전술한 바와 같이, 도 6은 본 발명에 따른 캐시 코히런시 모듈(111)에 Target address의 코히런시 유지 요청을 하는 함수에 대한 흐름도이다.

[0060] 도 6을 참조하면, Target address의 코히런시 유지 요청을 하는 함수는 먼저 Target address를 설정한 후(S510), 코드가 동작하는 코어의 NS bit가 1인지 확인한다(S511). NS bit가 0이면 Mode bit를 0으로 설정하고(S512) NS bit가 1이면 Mode bit를 1로 설정한다(S513). 설정이 완료되면, Enable bit를 1로 설정하여 캐시 코히런시 모듈(111)을 동작시킨다(S514). Enable bit는 캐시 코히런시 모듈(111)의 동작이 정상적으로 완료되면 0으로 설정된다. 그 후, 캐시 코히런시 모듈(111)을 동작시킨 후, Enable bit를 읽는다(S515). Enable bit가 1이면 S515 단계로 돌아가 다시 Enable bit를 읽고, Enable bit가 0이면 정상적으로 함수를 종료한다.

[0061] 최종적으로 본 특허를 통해 제안하는 AXI 프로토콜을 따라 동작하는 캐시 코히런시 모듈(111)을 이용하여 사용자는 캐시 flush 없이 다른 보안 모드로 데이터 전달이 가능하다.

[0062] 해당 함수는 동작중인 코어가 다른 모드로 전환하는 과정에서 빠른 속도를 위해 캐시 전체에 대한 flush 없이 모드 전환 후 데이터를 활용하려는 상황에서 사용할 수 있다.

[0063] 이상에서는 본 발명에 따른 FPGA 기반 캐시 코히런시 방법 및 이를 수행하는 장치에 대해 설명하였다. 이하에서는 본 발명에서 청구하고자 하는 FFPGA 기반 캐시 코히런시 방법을 수행하는 장치에 대해 설명한다. 이와 관련하여, 본 발명에서 청구하고자 하는 FFPGA 기반 캐시 코히런시 방법의 이하의 설명에 한정되는 것은 아니고, 전술한 모든 기술적 특징들을 적용하여 응용에 따라 변경 가능하다.

[0064] 이와 관련하여, 도 1 내지 도 5를 참조하면, FPGA 기반 캐시 코히런스 기능을 수행하는 장치는 프로세싱 시스템(100) 및 캐시 코히런시 모듈(111)을 포함할 수 있다.

[0065] 프로세싱 시스템(100)은 AMBA (Advanced Microcontroller Bus Architecture)에 따른 AMBA 송신 신호를 생성하여 송신하도로 구성된다. 코히런시 모듈(111)은 AMBA 송신 신호를 수신하면, 캐시 코히런시를 위한 AMBA 수신 신호를 프로세싱 시스템(110)으로 송신하도록 구성될 수 있다.

[0066] 프로세싱 시스템(100)은 캐시 코히런시 모듈(111)과 동작 가능하게 결합될 수 있다. 프로세싱 시스템(100)은 PL AXI 신호가 수신되면 코히런시 유지 요청을 하는 함수를 통해 타겟 주소와 모드 비트를 설정할 수 있다. 프로세싱 시스템(100)은 인에이블 비트를 1로 설정하여 캐시 코히런시 모듈(111)을 동작시킬 수 있다.

[0067] 캐시 코히런시 모듈(111)은 인에이블 비트(Enable bit) 값이 1로 설정되면, 타겟 주소의 읽기(read)에 대한 PL AXI 신호를 전송하고, 상기 PL AXI 신호에 대한 응답을 수신하도록 구성될 수 있다. 프로세싱 시스템(100)은 캐시 코히런시 모듈(111)의 동작이 정상적으로 완료되면 상기 인에이블 비트를 0으로 설정할 수 있다.

[0068] 캐시 코히런시 모듈(111)은 상기 응답이 프로세싱 시스템(100)으로부터 수신되면 타겟 주소의 쓰기(write)를 위한 제2 PL AXI 신호를 프로세싱 시스템(100)으로 전송할 수 있다. 이와 관련하여, 타겟 주소는 캐시 코히런시를 유지하기 위한 주소를 담아 저장하도록 구성될 수 있다.

[0069] 캐시 코히런시 모듈(111)은 모드 비트에 기반하여 연속되는 읽기 및 쓰기 동작에 대한 AxPROT 값을 결정할 수 있다. 캐시 코히런시 모듈(111)은 모드 비트가 1로 설정된 경우 Non-Secure 모드로 읽기 후 Secure 모드로 쓰기 동작을 수행할 수 있다. 반면에, 캐시 코히런시 모듈(111)은 모드 비트가 0으로 설정된 경우 Secure 모드로 읽기 후 Non-Secure 모드로 쓰기 동작을 수행할 수 있다.

[0070] 프로세싱 시스템(100)은 다른 캐시 또는 메모리보다 빠른 데이터 처리를 위한 고유의 L1 cache(102)를 가지고 있는 적어도 하나의 코어들(101)을 포함할 수 있다. 프로세싱 시스템(100)은 상기 코어들(101)과 동작 가능하게 결합되고, 상기 코어들(101)이 공유하도록 구성된 L2 cache(104)를 더 포함할 수 있다. 프로세싱 시스템(100)은

상기 코어 및 상기 L2 cache와 동작 가능하게 결합되고, 상기 코어들의 각 L1 cache와 상기 코어들이 공유하는 상기 L2 cache 간의 데이터 일관성을 유지하도록 구성된 SCU(103)를 더 포함할 수 있다.

[0071] FPGA 기반 캐시 코히런스 기능을 수행하는 장치는 메모리(120)를 더 포함할 수 있다. 메모리(120)는 L2 cache(104)와 동작 가능하게 결합되고, 정보를 저장하도록 구성된다. L2 cache(104)는 독점적으로 사용되는 상기 L1 cache(102)와 달리 상기 코어들(101)이 공유하고, 상기 L2 cache(104)는 상기 L1 cache(102)보다 데이터를 읽고 쓰는 속도가 느리고, 상기 메모리(120)보다 데이터를 읽고 쓰는 속도가 빠르게 설정될 수 있다.

[0072] 캐시 코히런시 모듈(111)은 프로그래머블 로직(PL, 110)에 포함되도록 구성된다. 프로세싱 시스템(100)은 AXI Master Port(105) 및 ACP(106)를 더 포함하도로 구성될 수 있다.

[0073] AXI Master Port(105)는 PL(110)의 AXI Slave Port(112)로 AMBA AXI4 프로토콜에 따라 PS AXI 신호를 전송하도록 구성된다.

[0074] ACP(106)는 PL(110)의 AXI Master Port(113)로부터 PL AXI 신호를 수신하도록 구성된다. 이와 관련하여, ACP(106)는 AMBA AXI4 프로토콜을 따르며 SCU(103)에 연결되어, 캐시 코히런시 모듈(111)이 L1 cache(102) 및 L2 cache(104)의 데이터 일관성에 관여하도록 구성된다.

[0075] 프로그래머블 로직(PL, 110)은 캐시 코히런시 모듈(111), AXI Slave Port(112) 및 AXI Master Port(113)를 포함하도록 구성될 수 있다. AXI Slave Port(112) 는 캐시 코히런시 모듈(111)과 동작 가능하게 결합되고, 프로세싱 시스템(100)의 AXI Master Port(105)로부터 PS AXI 신호를 수신하도록 구성된다. AXI Master Port(113)는 캐시 코히런시 모듈(111)과 동작 가능하게 결합되고, 프로세싱 시스템(100)의 ACP(106)로 PL AXI 신호를 전송하도록 구성된다.

[0076] 이상에서는 본 발명에 따른 FPGA 기반 캐시 무효화 방법 및 이를 수행하는 장치에 대해 설명하였다. 본 발명에 따른 FPGA 기반 캐시 무효화 방법에 대한 기술이 적용될 수 있는 제품 및 서비스는 다음과 같다.

[0077] 1) Mobile/IoT/Automotive 등 Arm 기반 SoC 환경에서 TrustZone을 사용하는 애플리케이션 최적화, 2) TrustZone 환경에서 동작하는 Operating System 최적화 및 3) Cloud/Big Data 관련 서비스의 FPGA 기반 성능 최적화에 활용될 수 있다.

[0078] 한편, 제품/서비스의 일부분을 구성하는 요소기술로 1) AAXI4 신호 변경을 통한 Cache Management 기술에 활용 가능하다. 한편, 2) Arm TrustZone 환경에서 동작하는 보안 기술에 활용 가능하다. 또한, 3) Arm TrustZone 환경에서 멀티코어를 활용하는 애플리케이션에서 활용 가능하다.

[0079] 본 발명에 따른 기술이 적용될 수 있는 시장은 1) 성능 및 보안에 민감한 Mobile, IoT, Automotive 시장 및 2) Arm FPGA를 적극적으로 활용하고 있는 Server 및 Cloud 서비스 시장일 수 있다.

[0080] 또한, 발명에 따른 기술을 활용할 수 있는 기업은 FPGA SoC 제조사 (Intel Altera, Xilinx 등) 및 FPGA 활용 기업 (Amazon, Google 등)과 Arm SoC 기반 애플리케이션 및 프레임워크 개발사 (스마트폰, 자동차 등)를 포함할 수 있다.

[0081] 본 발명에 따른 FPGA 기반 캐시 무효화 방법 및 이를 수행하는 장치의 기술적 효과는 다음과 같다.

[0082] 본 발명은 AXI ACP와 AMBA 통신규약을 이용하여 Armv8 CPU에서 경량화된 data coherency 유지 역할을 제공할 수 있다.

[0083] 본 발명을 통하여 정보보호를 보장하면서 빠른 동작 수행을 해야하는 프로그램을 효과적으로 설계할 수 있는 환경을 제공한다.

[0084] 본 발명의 IP를 이용하여, 사용자는 함수처럼 간편하게 data coherency를 유지할 수 있다.

[0085] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다.

[0086] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는바, 특정 실시 예들을 도면에 예시하고 상세한 설명에 구체적으로 설명하고자 한다. 그러나 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0087] 소프트웨어적인 구현에 의하면, 본 명세서에서 설명되는 절차 및 기능뿐만 아니라 각각의 구성 요소들에 대한 설계 및 파라미터 최적화는 별도의 소프트웨어 모듈로도 구현될 수 있다. 적절한 프로그램 언어로 쓰여진 소프트웨어 어플리케이션으로 소프트웨어 코드가 구현될 수 있다. 상기 소프트웨어 코드는 메모리에 저장되고, 제어부(controller) 또는 프로세서(processor)에 의해 실행될 수 있다.

부호의 설명

[0088] 100 : 프로세싱 시스템(Processing System: PS)

101 : Armv8 core

102 : L1 Cache

103 : 스누프 제어부(Snoop Control Unit: SCU)

104 : L2 Cache

105 : AXI (Advanced eXtensible Interface) Master Port

106 : ACP (Accelerator Coherency Port)

110 : 프로그래머블 로직 (Programmable Logic: PL)

111 : 캐시 코히런시 모듈

112 : AXI Slave Port

113 : AXI Master Port

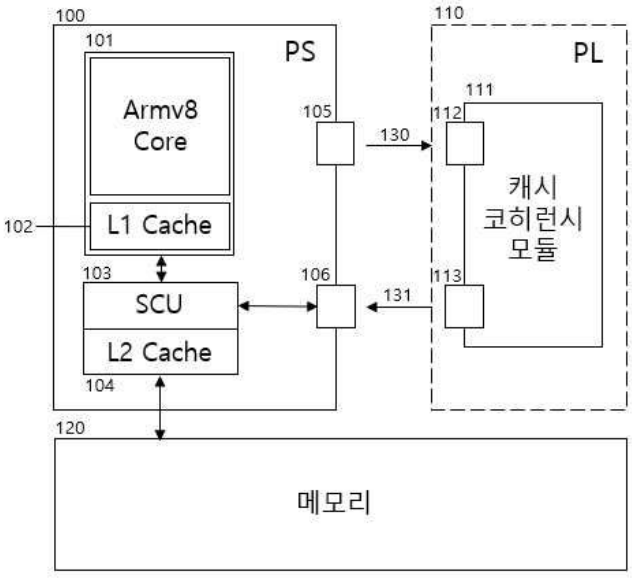
120 : 메모리

130 : PS AXI 신호

131 : PL AXI 신호

도면

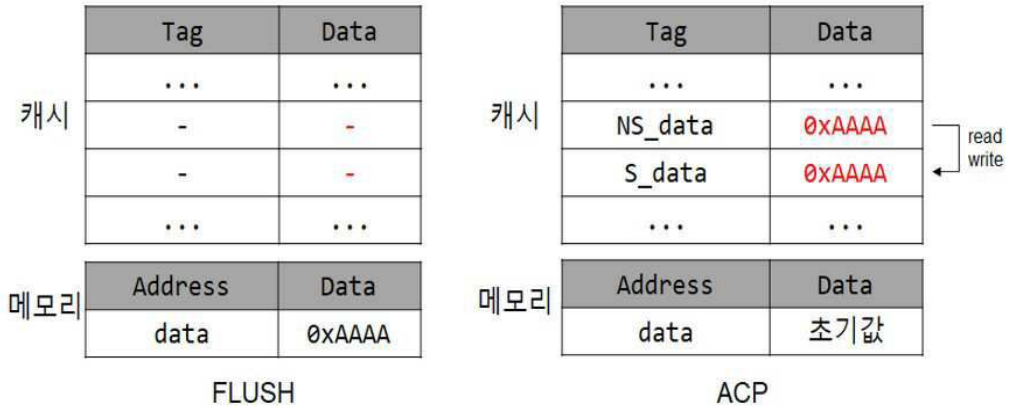
도면1



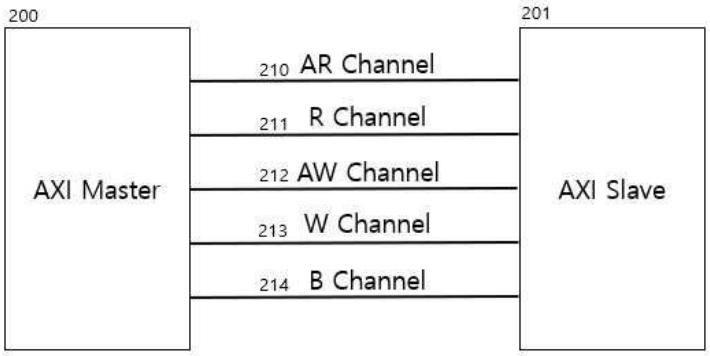
도면2a



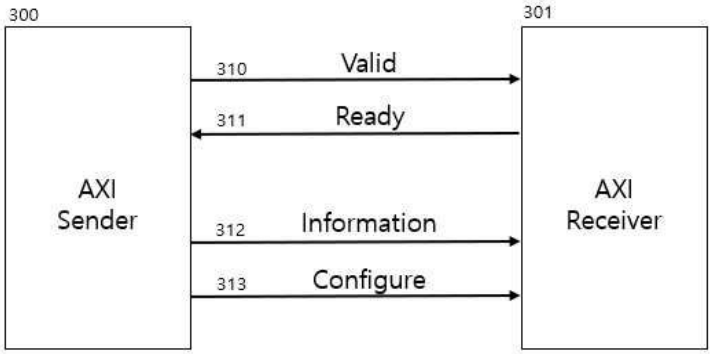
도면2b



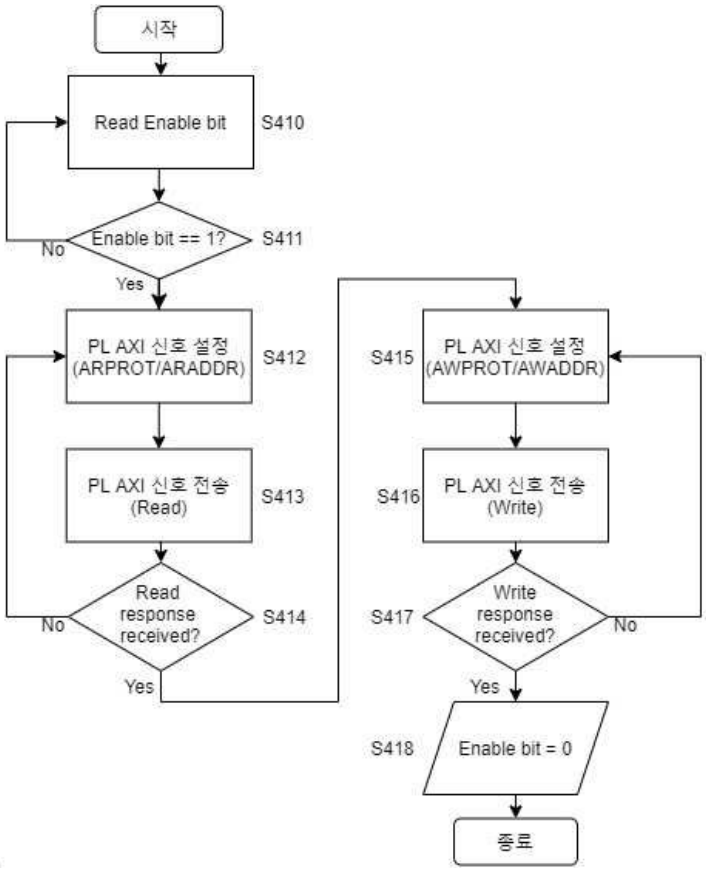
도면3



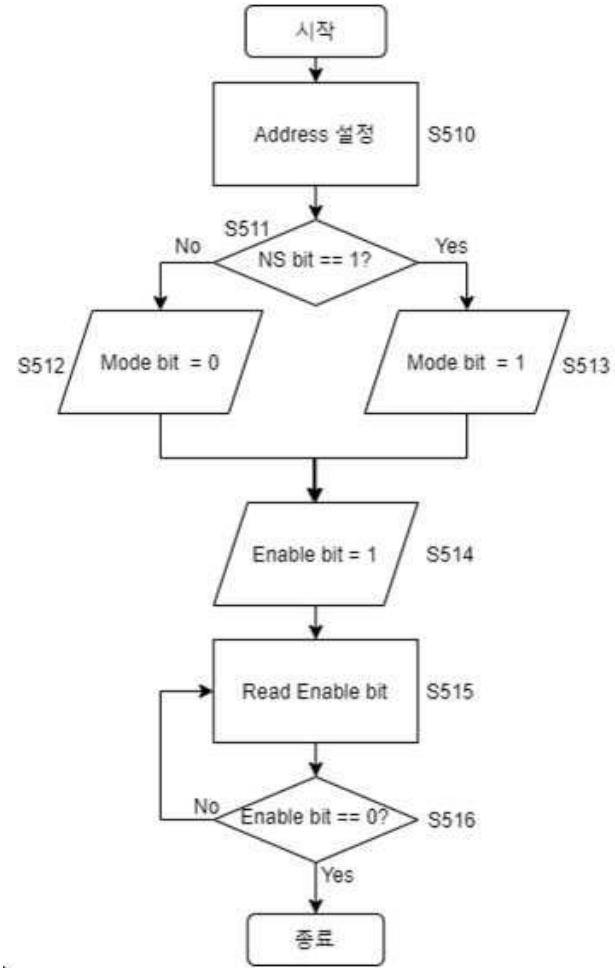
도면4


300
301
310
Valid
311
Ready
AXI
Sender
AXI
Receiver
312
Information
313
Configure

도면5


시작
Read Enable bit
S410
Enable bit == 1?
S411
No
Yes
PL AXI 신호 설정
(ARPROT/ARADDR)
S412
PL AXI 신호 전송
(Read)
S413
Read
response
received?
S414
No
Yes
S415
PL AXI 신호 설정
(AWPROT/AWADDR)
S416
PL AXI 신호 전송
(Write)
S417
Write
response
received?
No
Yes
S418
Enable bit = 0
종료

도면6


시작
Address 설정
S510
S511
NS bit == 1?
No
Yes
S512
Mode bit = 0
Mode bit = 1
S513
Enable bit = 1
S514
Read Enable bit
S515
Enable bit == 0?
No
S516
Yes
종료