

SYSTEM VERILOG PROJECT

VGA PROJECT

FPGA 이용한 실시간 동적 표적 추적 및 **LFSR**
기반 보안 영상 통신 시스템 구현

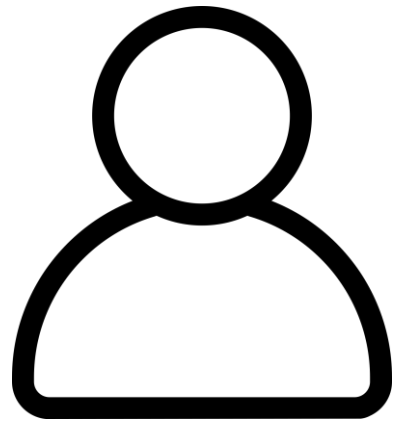
발표일 : 2025.12.05

발표자 : 박성훈

팀원 : 김승훈, 이상준, 조민준

Team

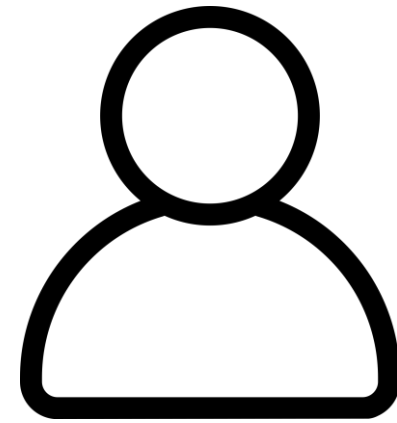
팀장
박성훈



담당 역할

- Top Integration
- Motion Display 설계
- PPT 작성
- 발표

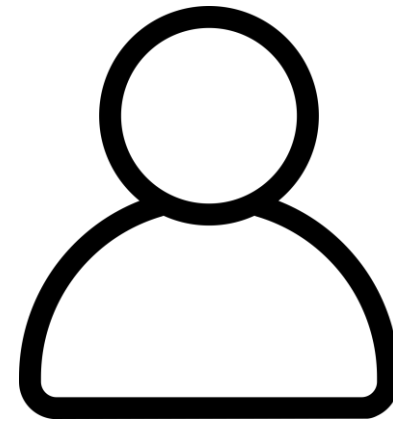
팀원
김승훈



담당 역할

- Dual Frame Buffer 설계
- Motion Compare 설계
- PPT 작성

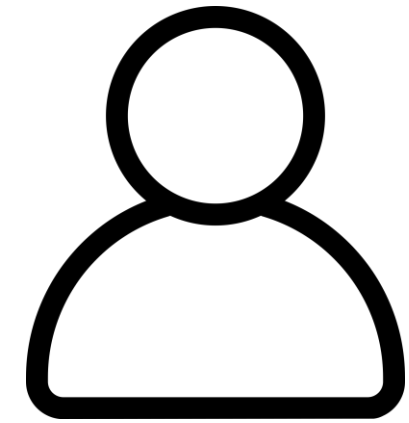
팀원
이상준



담당 역할

- Motion Display 설계
- LFSR 설계
- PPT 작성

팀원
조민준



담당 역할

- SCCB 구현
- 동작 영상 촬영
- PPT 작성

Contents

01

Introduction

02

System Architecture

03

Function

04

동작영상

05

Trouble Shooting

06

Conclusion

01 Introduction

Introduction



미래는?

- UAM(Urban Air Mobility)
- Military Drone



Introduction

✓ 경찰 드론이 지휘소로 영상데이터 송신



✓ Check Point

- 움직이는 물체를 어떻게 포착할 것인가?
- 드론이 찍은 영상은 그대로 송신해도 되는가?
- 암호화를 한다면 받는 곳에서는 복호화를 어떻게 해야하는가?

Introduction

✓ 프로젝트 목표

- FPGA 하드웨어 가속을 이용한 실시간 동적 표적 추적 및 LFSR 기반 보안 영상 통신 시스템 구현

✓ 핵심 기능

- 실시간 영상 수집 및 모션 감지 (Real-time Motion Detection)
- 동적 표적 추적 알고리즘 (Dynamic Target Tracking)
- LFSR 기반 하드웨어 암호화 (Hardware Encryption Via LFSR)
- 동기화된 복호화 및 원격 모니터링 (Synchronized Decryption & Monitoring)
- 표적지 탄환 발사

구현환경

✓ Equipment

AMD



- Basys3
- OV7670

✓ Language



- System Verilog

✓ Design Tool



- Vivado

✓ Code Management

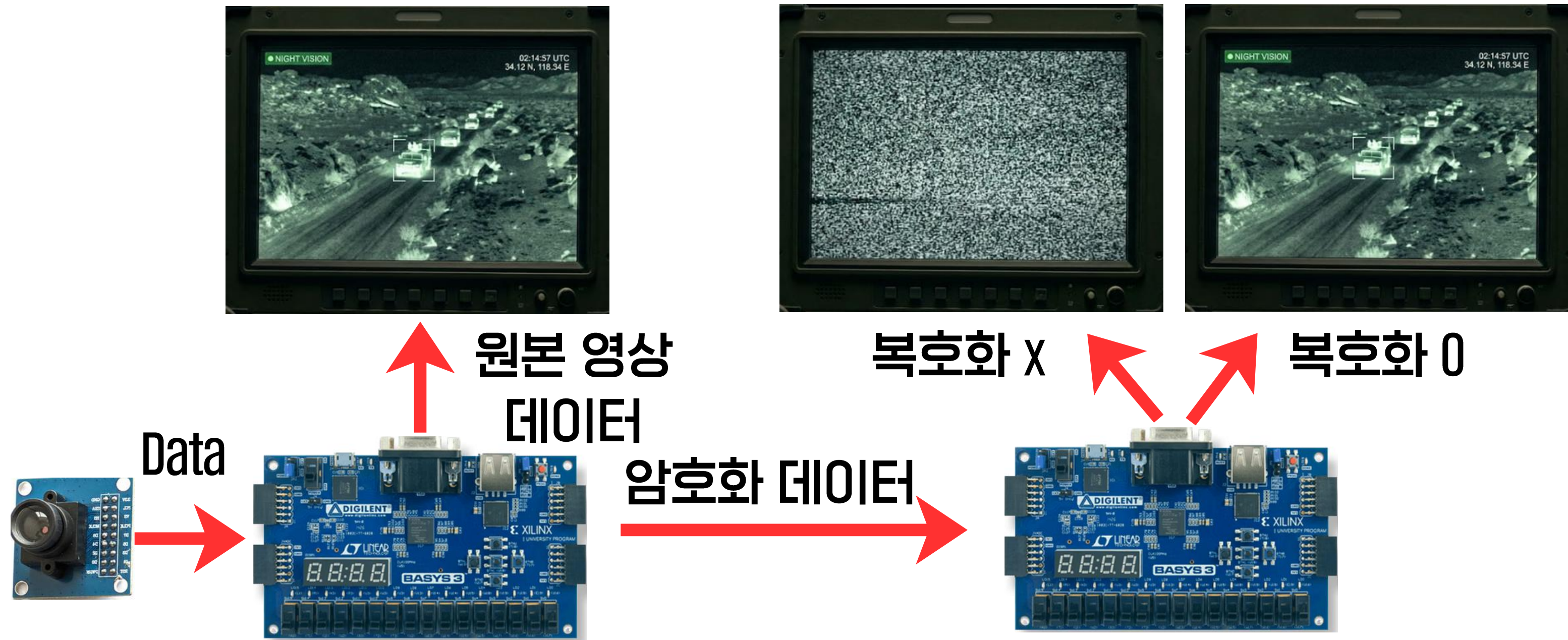


- Git
- GitHub

02 System Architecture

System Architecture

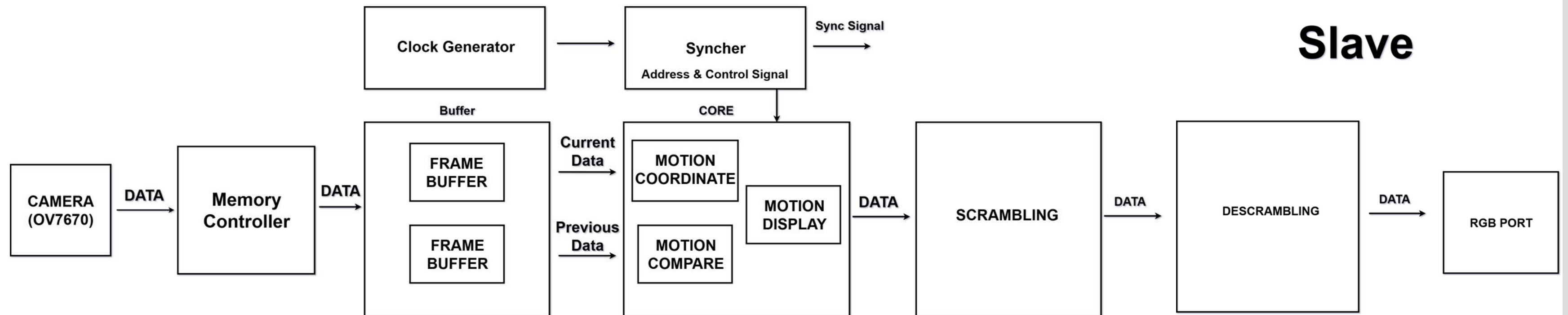
✓ Schematic Diagram



System Architecture

✓ Block Diagram

Master



Slave

✓ MASTER

- Buffer
- Core
- Scrambling

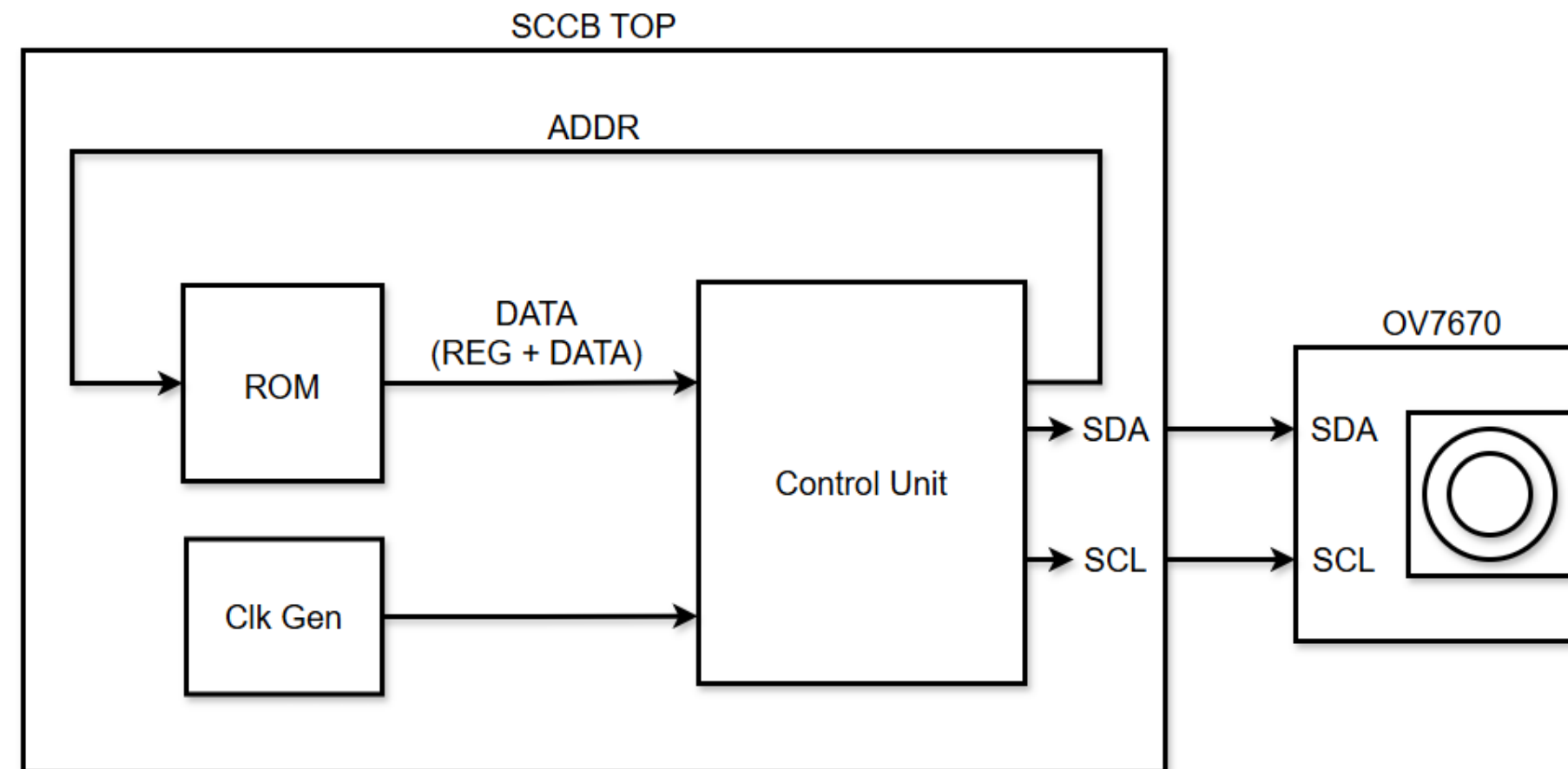
✓ SLAVE

- Descrambling

03 Function

Function

✓ SCCB Master

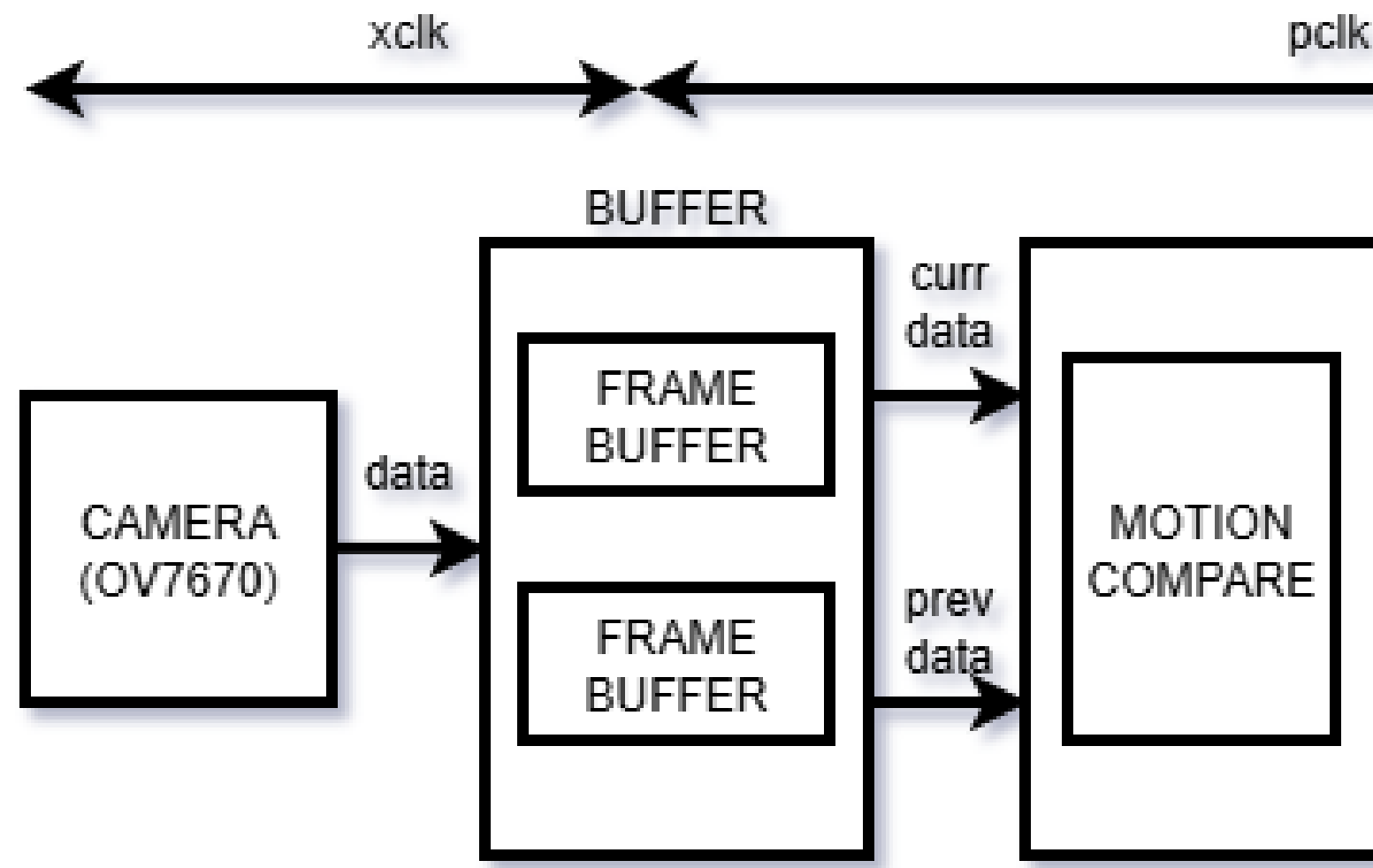


✓ Check Point

- SDA/SCL신호를 이용해 카메라를 제어하는 모듈
- ROM 데이터를 Read 후 카메라의 레지스터 값을 수정
- 카메라(OV7670) SCL 타이밍을 생성을 위한 CLK_GEN을 사용

Function

✓ Buffer Logic

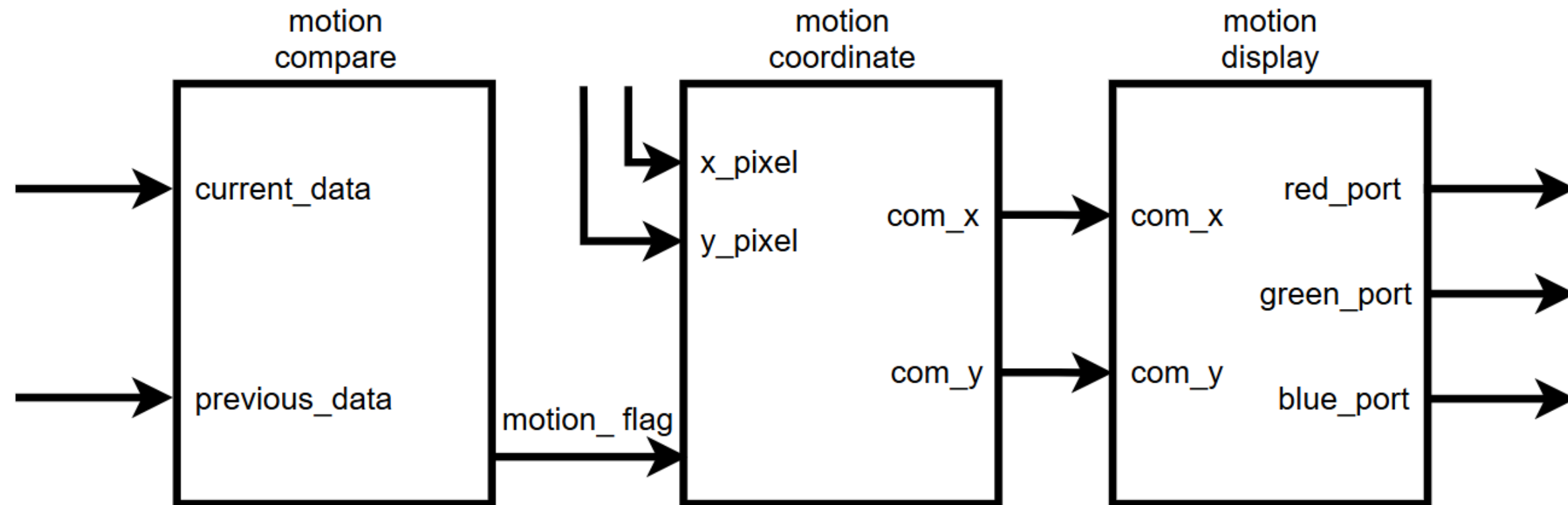


✓ Check Point

- 카메라에서 들어오는 한 프레임을 메모리 저장
- xclk과 pclk → Write, Read 동작을 분리
- 모션 검출 : 이전 프레임과 현재 프레임 밝기 비교
- Ping-pong Buffering 구조 구현

Function

✓ Motion Compare & Motion Coordinate & Motion Display



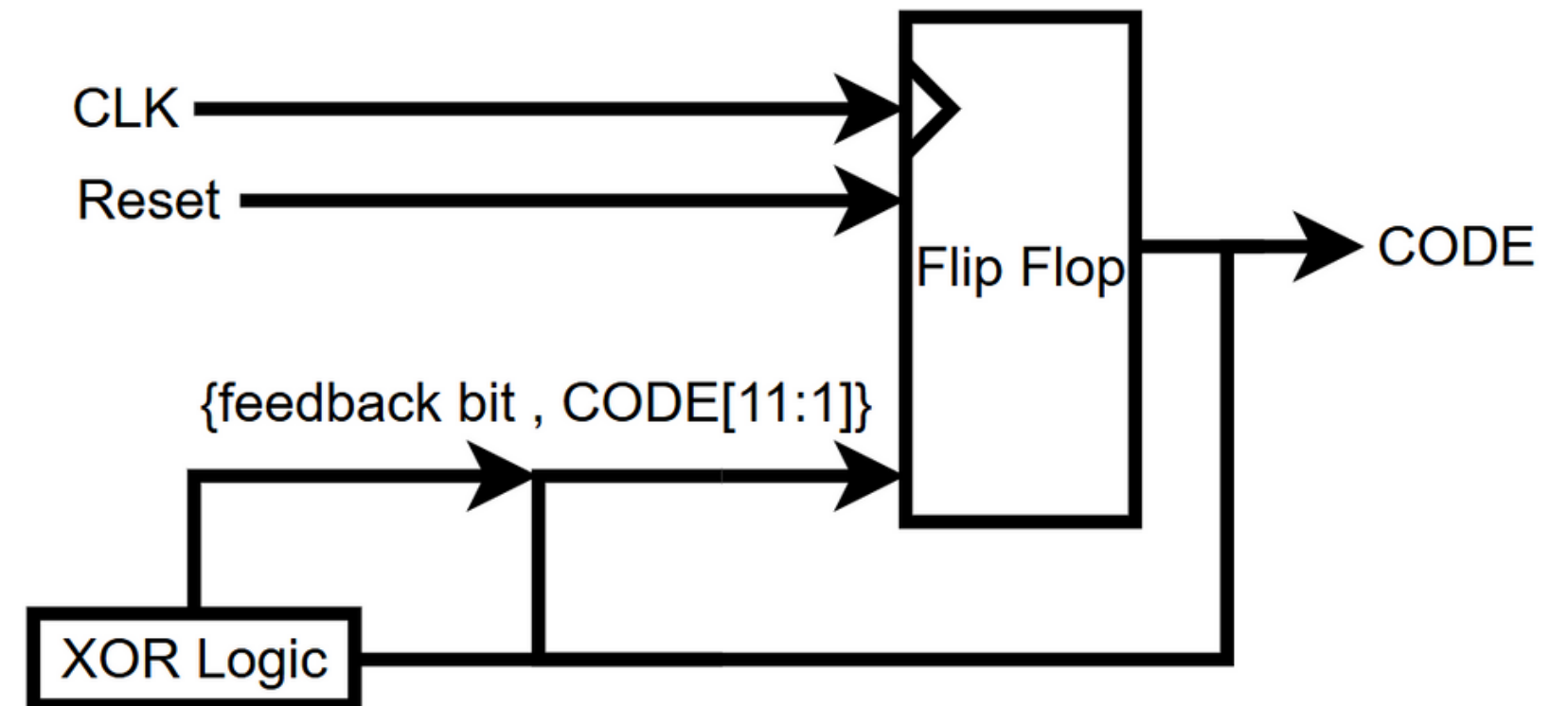
✓ check point

- Motion Compare : RGB-to-Grayscale 변환 및 Frame Difference(차분) 기반 모션 감지
- Motion Coordinate : 활성 픽셀(Flag) 누적 연산을 통한 표적 중심 좌표(Center of Coordinate) 산출
- Motion Display : 좌표 기반 표적 추적 박스(Target Indicator) 오버레이 및 영상 합성

Function

✓ LFSR(Linear Feedback Shift Register)

원본 데이터	Key가 0일 때	Key	Result	
1	XOR	0	1	Identity
0	XOR	0	0	Identity
원본 데이터	Key가 1일 때	Key		
1	XOR	1	0	Inverter
0	XOR	1	1	Inverter



$$\text{feedback bit} = \text{CODE}[11] \wedge \text{CODE}[5] \wedge \text{CODE}[3] \wedge \text{CODE}[1]$$

✓ check point

- Master : 원본 픽셀 데이터 마다 XOR 암호키 값 생성
- Slave : Master에서 생성하는 암호키와 동일한 타이밍에 생성
- 암호화 원리 : Scrambled data = 원본 data \oplus 암호키
- 복호화 원리 : 원본 data = Scrambled data \oplus 암호키

04 동작영상

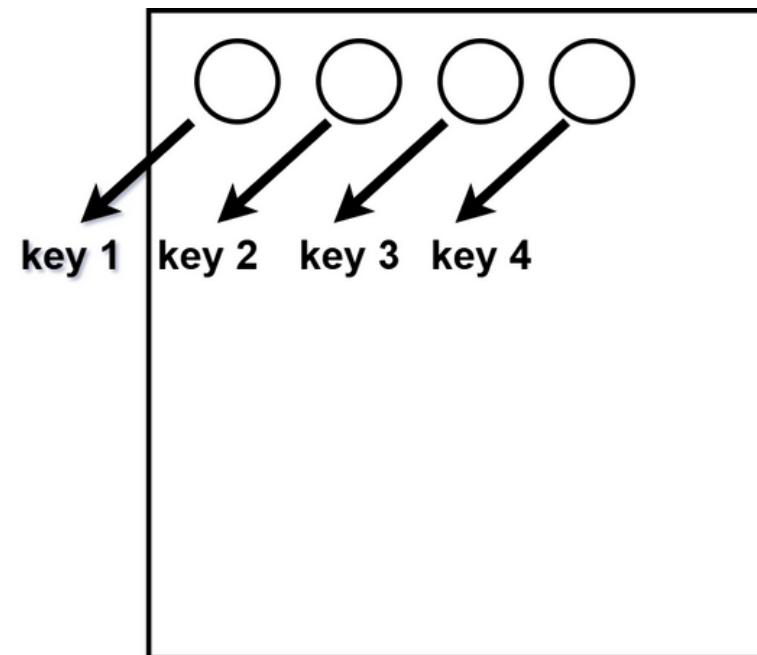
동작영상



05 Trouble Shooting

Trouble Shooting

✓ Case : 암호화 과정



Scrambled data = 원본 data \oplus 암호키

원본 data = Scrambled data \oplus 암호키

문제상황 : 고정된 12bit CODE로 모든 픽셀 암호화 시 색만 바뀌는 현상

해결방법 : 12bit CODE 값을 고정값이 아닌 LSFR 로직을 통해 매 클럭마다 CODE를 바꿔주는 로직으로 생성



문제상황 : 암호화 과정 (LFSR) 에서 사용한 클럭을 100MHz 을 이용하여서 적은 경우의 수(1/4) 사용

해결방법 : 100MHz 클럭이 아닌 25MHz PCLK 을 이용.

Trouble Shooting

Case : Port Mapping Rule Violation

문제상황 : Master → VGA 전송 데이터 정상 출력, Master → SLAVE 로 전송 데이터 출력 X

해결방법 : 데이터, 싱크중 어떤 것이 문제인지 알 수 없었고, ANALYZER를 이용하여 포트의 출력 신호 확인

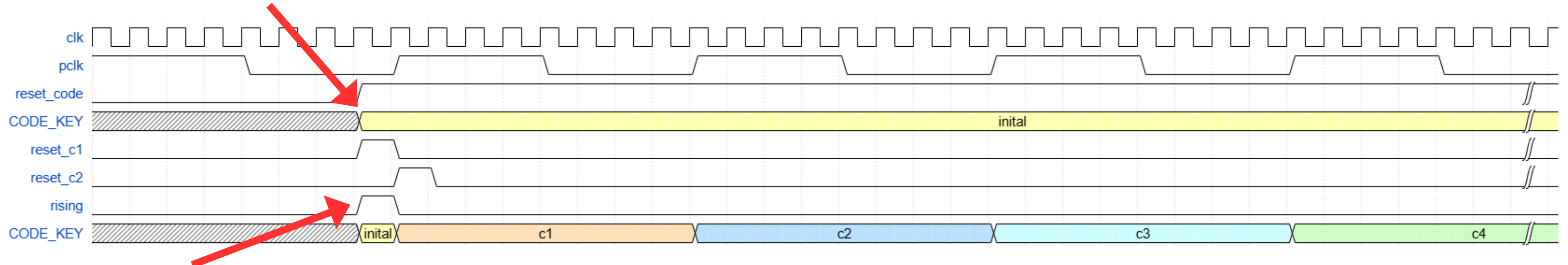
문제 원인 : Output Data는 동시에 여러 포트 동시 할당 (XDC 파일) 불가능

#결과 : 2개의 OUTPUT Port 연결, MASTER VGA와 SLAVE VGA 모두 정상적으로 동작 확인.

Trouble Shooting

✓ Case : 복호화 과정

초기 code값 유지



Rising Signal(10ns) 생성

문제상황 : Reset_Code일 때 CODE 가 초기 값으로 고정되어 암호화 X

문제정의 : KEY 값이 틀리더라도 CODE값이 일정하여 색만 바뀌는 노이즈로 변경

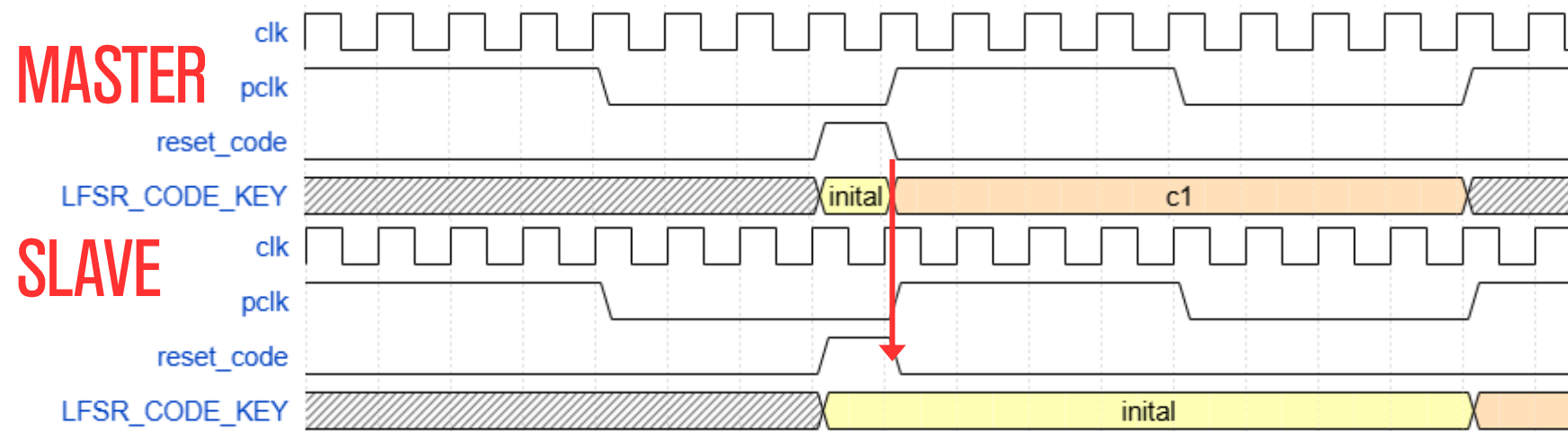
해결방법 : Reset의 Rising Edge(10ns) Signal 생성

결과 : 리셋 버튼을 꾹 누르고 있더라도 Rising Edge 에서만 reset 이 되어 원하는 암호화가 가능.

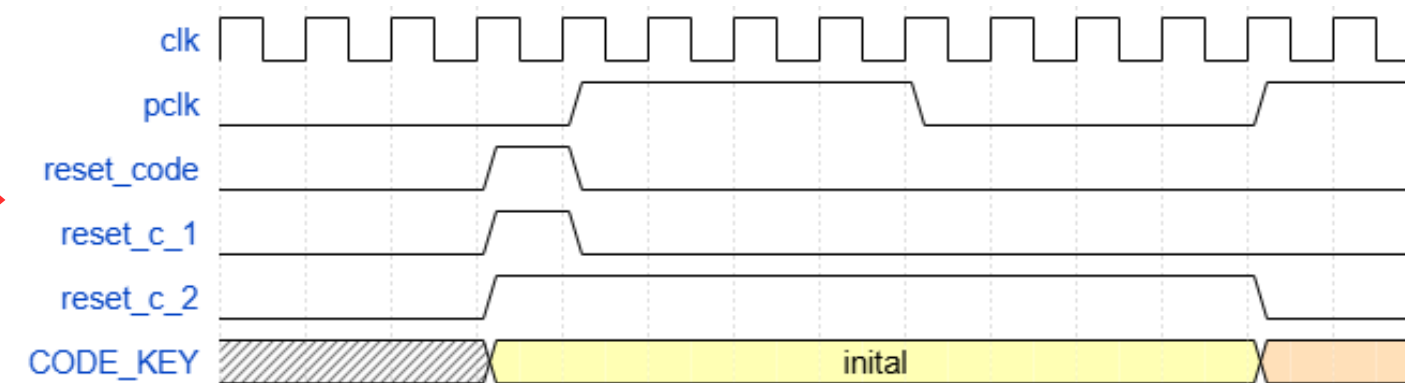
Trouble Shooting

✓ Case : 암호화 키 초기화 동기 실패 원인 분석 및 신호 무결성(Signal Integrity) 확보

✓ Before : Timing Diagram



✓ After : Timing Diagram



문제상황 : Reset 버튼을 눌렀을 경우 암호키 동시 Reset 실패

문제정의 : 10ns rising 신호가 posedge pclk에 falling했을 시에 Metastability

해결방법 : Reset의 Rising Edge를 40ns 이상 길게 유지

결과 : 암호화, 복호화 모듈 모두 동일한 타이밍에 초기화

Trouble Shooting

✓ Case : 움직임이 없을 때 LOCK ON 마크가 남아있음

문제상황 : 움직이는 물체가 화면에서 사라졌는데도, 이전에 찍힌 LOCK ON 계속 남아 있음

문제정의 : 움직이는 물체가 없으면 LOCK ON 마크가 없어야 정상, 그렇지만 이전 프레임 중심 좌표 값이 유지되어 마지막 위치에 마크가 고정되는 문제 발생

해결방법 :

1. Motion_coordinate(모션 중심 좌표 계산)에서 중심 좌표값 초기화
2. motion_valid(프레임 단위)를 추가하여 움직임이 감지되면 1, 아니면 0을 motion_display로 출력
3. 움직임이 감지되지 않았을 때는 원본 RGB 픽셀값이 출력되게 함

결과

움직이는 물체가 없을 경우 LOCK ON 마크가 사라지고 원본 카메라 영상만 출력되는 것을 확인

06 Conclusion

Discussion

✓ Dual Frame Buffer를 왜 사용했는가?

- 설계 의도: 실시간으로 움직이는 물체를 추적하기 위해 Frame Difference Algorithm 채택
- 조명 변화 및 노이즈에 대한 민감도 문제
- Multi Object Tracking 제한

✓ 영상 데이터 Noise

- 설계 의도: SCCB 모듈로 카메라 레지스터를 설정해 QQVGA 영상 출력
- 좌측 가장자리 라인에서 잡음 발생
- Frame Buffer에서 가장자리의 주소를 read하는것을 제한

Conclusion

✓ 프로젝트 성과

- OV7670 카메라 제어 및 Ping-Pong Buffer 구현
- 프레임 간 차분 기법을 통해 실시간 동적 표적 추적기 구현
- LFSR을 이용해 실시간 영상 암호화 및 복호화 구조 구현

✓ 향후 계획 및 배운 점

- 현재 좌표 중심 계산 방식은 여러 물체가 움직일 때 추적 제한 → 추후 Multi-Object Tracking으로 발전
- 각기 다른 보드에서 보내는 신호에 대한 디버깅에는 Analyzer 필수
- LFSR 구현하면서 Analyzer 및 Test Port를 이용한 Debugging Technique 향상 및 Metastability Timing 이해도 향상

소감

팀장 박성훈

프로젝트를 진행하면서 협업의 방식, 데이터 Timing의 중요성, Metastability 원인 파악과 Debugging의 여러 방식을 배웠습니다.

Git으로 코드 협업 효율을 올리고 두 보드간 데이터 동기화 구현과 Analyzer를 통해 디버깅을 하며 기능 구현을 했을 때 많은 것에 성장했다고 생각하고 큰 성취감을 느꼈습니다.

팀원 김승훈

설계를 진행하면서 BRAM 자원 사용량을 고려한 메모리 구조 설계의 중요성, 디버깅을 통해 CDC, VGA timing에 대해 많은 것을 배울 수 있었습니다.

협업하는 과정에서 코드 관리 방식과 모듈간 인터페이스를 명확히 정의하는 것이 중요하다는 것을 느꼈고, 이를 통해 디버깅과 유지보수가 훨씬 수월해진다는 것을 깨달았습니다.

팀원 이상준

문제가 생겼을 때, 해결하는 과정에서 서로 의견을 나누고, 성공하였을 때 큰 성취감과 즐거움을 느꼈습니다. 협업을 하는 과정에서 소통의 중요성을 다시 배우게 되었습니다.

그동안 도와준 팀원들 모두 고마웠고 앞으로도 행복하게 잘 살았으면 좋겠습니다.

팀원 조민준

SCCB를 설계를 하며 I2C 통신 프로토콜에 대해 부족했던 부분을 채울 수 있는 시간이 되었습니다.

또한, 협업을 하는 과정에서 서로 어떤 목표로 기능을 구현해야할 지 의논하고 진행상황을 공유하는 과정등을 통해서 소통의 중요성을 느꼈습니다.

감사합니다