

실시간 화면 분할 시스템 구현에 관한 연구

백 철^o • 박 인 규

홍익대학교 대학원 전기제어공학과
121-791 서울 마포구 상수동 72-1
gwhitefe@wow.hongik.ac.kr

A study on implementing of real time multi-viewer system

Cheul Paik^o • In-gyu Park

Dept. of Electrical & Control Eng.,
Graduate School, Hong Ik University
72-1 Sangsu-dong Mapo-gu Seoul 121-791
gwhitefe@wow.hongik.ac.kr

요약

현재 보안 시스템으로 가장 많이 쓰이고 있는 것 중에 하나가 여러 지역의 카메라로부터 영상 신호를 받아서 하나의 모니터에 여러 영상을 분할해서 보여주는 시스템이다. 이 시스템의 기능 중에서 가장 중요한 것은 각 지역의 영상을 실시간으로 처리해줄 수 있어야 하는데, 이를 위해서는 영상 데이터를 놓치지 않고 모두 메모리에 저장할 수 있어야 한다. 본 논문에서는 4개의 영상을 하나의 화면으로 분할하여 출력하기 위한 시스템을 FPGA를 사용하여 구현했다.

일반적으로 화면 분할하는 시스템은 흑백의 영상만을 출력하는데, 컬러 영상 신호인 RGB 5:6:5모드의 데이터를 사용하여 컬러 영상을 그대로 화면 분할하여 출력하는 시스템을 구성했다. 또한, 화면을 나누기 위한 PIP(Picture In Picture) 등의 전용칩은 분할 화면의 수가 늘어날수록 그 시스템의 크기가 키지므로 순수하게 FPGA를 이용하여 로직을 설계해서 직접 램드 메모리(FIFO)를 컨트롤 하도록 설계했다.

동기화 되어 있지 않은 메모리에 저장한 각 영상 데이터를 하나의 영상화면에 동기화시키기 위한 방법으로 일정한 타이밍마다 각 영상 데이터를 선택하는 선택 알고리즘(Choice Algorithm)을 제시하여 적용하였다.

선택 알고리즘에 따라서 동기화 되어 있지 않은 메모리에 저장한 각 영상 데이터를 하나의 영상화면에 동기화시키기 위한 방법을 로직으로 구현하여 적용한 시스템을 만들어서 직접 실험 및 테스트를 실행하였다. 로직을 구현하기 위해 사용한 FPGA(Xilinx 5200 Series)는 XC5210-5이고, 비디오 데이터를 저장하기 위한 램드 메모리(FIFO)는 μ PD42280-30를 사용하였는데, 좀더 여유 있는 데이터 저장을 통해 선명한 화질을 얻기 위해서는 FPGA와 메모리를 더 빠른 타입으로 사용하는 것이 바람직하다.

내용 전개를 살펴보면 제 1절에서 본 시스템의 필요

성 및 개발 동기, 개발 배경 등에 대해서 간단히 설명하고 제 2절에서는 전체 시스템의 구조에 대해서 설명하고 제 3절에서는 본 시스템의 구조 중에서 가장 중요한 메모리 컨트롤에 대해서 간단히 설명하고, 제 4절에서는 시스템을 구현시켜 실험 및 결과에 대해서 분석한다. 마지막으로 결론 및 향후 계획에 대해서 기술한다.

1. 서론

현대의 급속한 산업발전에 의한 주거지와 상업 지역 등의 분리는 사람이 일일이 감시할 수 없는 지역에 대한 보안의 중요성이 점점 강조되고 있다. 심지어는 전문적으로 보안을 담당하는 회사들이 많이 생기고 있다. 그러나, 실제 이러한 보안 회사들이 사용하는 장비들은 거의가 외국의 고가 장비를 그대로 수입해서 사용하고 있는 실정이다.

그 중에 움직임이나 외부 접촉 등에 반응하는 센서 등의 부분은 기초적인 소자 연구 개발이 이루어져야 가능하지만, 영상이나 통신 쪽의 장비들은 충분히 국내의 기술로 개발 가능하다고 생각한다. 특히 원격지는 급속하기 위하여, 직접 영상을 전송 받아서 감시국에서 이를 모두 관리하는 시스템의 경우, 각 원격지의 영상을 쉽게 관리하고 감시할 수 있도록 분할해서 하나의 영상으로 보여주는 것이 중요하다. 이러한 장비는 꼭 보안 전문회사가 아니더라도 일반 상점 등에 사용할 수 있는 장비로 현재는 일본 등지에서 거의 수입해서 사용하고 있는 실정이다.

이러한 영상 처리의 경우 일반 화상처리의 경우처럼 영상 데이터를 DCT나 그 외의 복잡한 이미지 프로세싱의 과정을 거치는 것이 아니라 입력된 영상을 실시간으로 원하는 위치와 크기로 바꾸어 주어 출력하는 것이 중요하다. 여러 영상을 분할해서 하나의 화면으로 출력하기 위한 방법으로 주로 사용하고 있는 방법은 전용 칩인 PIP(Picture In Picture)를 이용해서 기존 화면에 다른

영상 테이터를 삽입하는 형식이다.

위의 방법은 전용 칩을 사용하기 때문에 화면 분할의 구현이 용이하나, 2개의 화면을 중첩시키기 위해서는 MIP 전용칩, 필드 메모리, 그리고 영상 신호를 위한 A/D(Analog to Digital Converter), D/A(Digital to Analog Converter)가 필요하다. 즉, 화면 분할의 수가 늘어 날 수록 중첩시켜야 되는 영상 신호가 늘어나기 때문에 이에 따른 부품의 증가와 비용의 증가가 생기게 된다.

이러한 단점을 보완하기 위해서 화면을 분할하는 부분을 FPGA를 사용해 분할하는 영상의 수가 많아질 경우에도 적용할 수 있도록 구성한다.

다종의 영상신호를 받아서 하나의 화면으로 분할하는 시스템에서 가장 중요한 것은 각 영상을 실시간으로 처리해줄 수 있어야 하는데, 이를 위해서는 각 입력되는 영상 테이터를 놓치지 않고 메모리에 저장한 후 동기를 맞추어서 화면에 디스플레이 할 수 있어야 한다.

본 논문에서는 여러 영상 테이터의 동기를 맞추고자 할 경우, 각각의 데이터 저장에 대한 메모리 제어를 해야 한다. 이를 위해서 각 영상 테이터를 저장할 수 있는 버퍼(Buffer)역할을 하는 필드 메모리를 2개의 뱅크(Bank)씩 두어 동기여 맞도록 선택할 수 있는 시스템을 구성한다. 이 시스템의 목적은 기존의 화면 분할 시스템에서 발생하는 분할 수의 증가에 따른 추가 부담을 줄이고, 수입에 의존하던 보안시스템의 장비 개발의 기술 축적에 있다.

2. 시스템 구조

2.1 시스템 보드의 사양

2.1.1 시스템 보드의 기능

4개의 영상을 외부로 부터 입력 받아서 하나의 화면으로 분할 해서 출력하는 시스템을 구성한다. 이 시스템을 보안 시스템으로 사용하기 위해서 다음과 같은 기능을 지니도록 설계한다.

◆ 외부 4개의 영상 신호에 대한 4분할 화면의 실시간 출력

① 4개의 영상 신호 입력 데이터 변환 : SAA7111*4 (NTSC → RGB 5:6:5)

② Field Memory : μ PD42280(FIFO, 8 bit, 30ns Write Time)

◆ 외부 모니터로 분할된 화면 출력

범용 적인 화면 출력을 위하여 여러 가지의 영상 출력 형식 지원.

① Interlace 주사 방식 (NTSC, RGB) 지원 :

→ B856(Video Encoder)

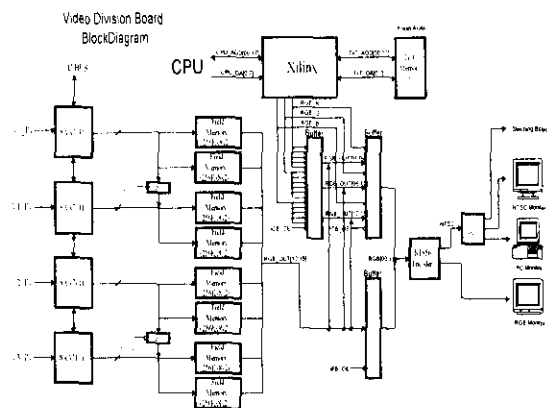
② Non-Interlace 주사 방식 (RGB) 지원 :

→ TV 수신카드.

2.2 보드의 블록 및 기능

앞서 제시한 보안 시스템으로 사용하기 위해서 필요한 기능들을 만족하는 보드를 설계하기 위해서는 먼저, 각 기능을 담당하는 블록을 설정하고 그 블록에서 사용되어질 구체적인 칩과 신호들을 정의해야 한다.

다음의 블록도는 전체 시스템을 요구하는 여러 기능 중심으로하여 도식적으로 표시한 것이다.



모든 전체의 블록도에서 각각의 블록에 대한 간단한 설명은 다음과 같다.

◆ SAA7111 비디오 디코더 칩으로 아날로그 영상 신호를 받아서 YUV 4:2:2 또는 RGB 5:6:5 로 디지털 데이터로 변환한다.

◆ Xilinx 4개의 영상 신호가 디코더(SAA7111)을 거친 클럭을 모두 받아서 선택 알고리즘 로직을 거쳐 메모리를 컨트롤 한다. 8051 CPU와 interface 해서 IC 버스를 컨트롤하고, SAA7111로부터 받은 4개 영상의 PIXCLK을 받아서 필드 메모리 컨트롤하고 더불어, 보드 내의 버스 버퍼들을 컨트롤 한다. 그리고 분할된 화면에 한글 텍스트 등을 넣기 위한 텍스트 메모리 제어하며 필드 메모리에서 읽은 데이터와 텍스트 메모리의 내용과 비교하며 RGB 16 bit Data를 출력한다.

Input - 1. 4개 영상에 대한 Sync & Clock

- 2. Video Encoder Sync & Clock
- 3. Text Memory Address & Data
- 4. CPU Address & Data

- Output ~
- 1. Field Memory Control
 - 2. Buffer Control
 - 3. Switching Control
 - 4. SAA7111 DATA Control
 - 5. Fresh Memory Control

◆ **Field memory** 입력되는 디지털 영상 데이터를 저장하는 메모리로 입력과 출력시에 서로 병행하면서 사용할 수 있도록 2개의 프레임을 저장 가능한 크기로 정한다. 메모리 크기와 모델은 다음과 같다.

(256K x 8bit) x 4, Model : μ PD42280/30

◆ **Text memory** 분할된 화면에 특정한 이미지나 한글을 표현할 수 있도록 한 메모리로 하나의 프레임에 해당하는 데이터를 저장하게 한다. 단, 픽셀 당 8 bit의 데이터를 할당한다.

256K x 8 bit, Model : AT49F002T 30

◆ **BI856** 디지털 데이터 RGB 5:6:5 형식의 16 bit를 받아서 NTSC, S-VHS는 물론이고 Analog RGB까지 출력한다.

3. 메모리 컨트롤 방법

3.1 메모리 컨트롤 방법

3.1.1 다중 영상 데이터 저장의 문제점

화면을 분할하기 위한 외부의 영상은 서로 다른 카메라를 통해서 입력되는데, 이때의 영상신호는 특별히 "GENLOCK"기능을 써서 동기화(Synchronize)시키지 않는 이상 서로 다른 영상동기(Video Sync)를 갖게 된다. 즉, 각 영상의 수평동기(Horizontal Sync)와 수직동기(Vertical Sync)의 주기나 위상이 서로 다르게 된다.

영상신호를 저장하기 위해서는 직접 아날로그 신호(여기서는 NTSC : National Television System Committee 사용)를 메모리에 저장하는 것이 불가능하기 때문에 영상 전용 디코더를 사용하여 디지털 데이터로 변환하는 과정이 필요하게 된다. 이렇게 변환된 디지털 데이터는 디코더의 해상도(수평 해상도는 NTSC의 경우는 525라인, SECAM일 경우에는 625라인으로 정해져 있으므로, 여기서 해상도는 수평 해상도를 말함)에 따라서 일정한 픽셀주기(Pixel Rate)를 갖게 되는데, 예를 들어서 수평 해상도가 720인 경우 수평동기의 주기는 약 63.5 μ s 이므로 픽셀 주기는 88ns 가 된다. 그러나, 실제 디코더에서 나오는 픽셀의 주기는 70~75ns

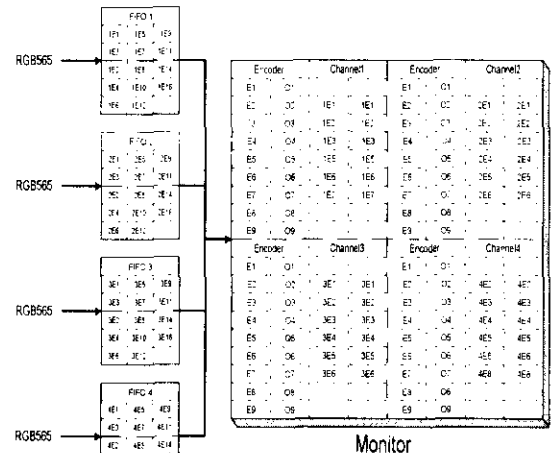
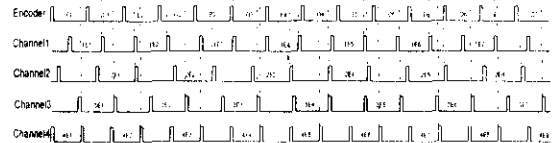
이며, 주기가 짧은 이유는 수평 주기에는 블랭크 신호, 버스트 신호등이 포함되어 있어 화면상에 유효한 시간이 63 μ s 보다 작기 때문이고, 주기가 가변적인 이유는 디코더의 내부에서 카메라의 영상신호의 동기 맞추어서 PLL(Phase Locked Loop)를 실행하기 때문이다. 즉, 디지털로 변환된 각 영상 데이터의 수직, 수평 동기는 물론이고 픽셀주기까지 서로 다르게 된다.

위와 같은 이유 때문에 2개 이상의 디지털 영상 데이터 하나의 메모리에 저장할 경우, 어느 하나의 영상 동기에 맞추어 저장하게 되면 다른 영상데이터는 손실될 수 있다. 그래서 각 영상마다 2개의 버스트를 두어 그 영상의 동기에 맞게 메모리에 저장한 후에 각각의 동기에 맞추어 저장된 영상 데이터를 하나의 동기에 일치시켜 디스플레이 시키는 방법을 알아 보고자 한다.

3.1.2 데이터 선택 알고리즘

2개 이상의 영상 데이터를 저장할 때 서로 다른 동기 때문에 발생하는 문제를 해결하기 위해서 각 영상마다 할당된 메모리로 저장한 후 인코더의 동기에 맞추어 아래의 그림과 같이 각 영상의 두 버스트 중 하나를 메모리를 선택하여 읽는다.

4분할 Timing Diagram



여기서 주의해야 할 것은 디코더쪽의 동기화 인코더쪽의 동기가 맞지 않으므로 인해서 발생하는 화면에 줄이

가는 현상이 나타나지 않도록 뱅크선택을 해야한다.

이 알고리즘의 방법을 다음과 같은 조건하에서 구체화시킨다.

[알고리즘 구현 조건]

- ◆ 아날로그 영상 신호의 규격은 NTSC
- ◆ 입력되는 영상은 4개이고, 신호는 서로 비동기
- ◆ 영상 데이터는 4분할을 위해 1/2로 축소 샘플링
- ◆ 한 데이터 픽셀주기는 70 ~ 75ns

데이터 우선 순위의 알고리즘을 구현하기 위해서 위의 조건과 같이 입력되는 영상의 수는 4개로 하고, 한 화면에 4분할하여 출력하도록 한다. 이러한 조건하에서 4개의 영상 데이터를 각각의 메모리에 저장한후 다음과 같은 방법에 따른다.

[알고리즘 구현 방법]

- ◆ 각 영상의 메모리의 두 뱅크중 어느 뱅크 어느 부분에 영상 데이터를 쓰고 있는지 체크함.
- ◆ 인코디에서 각 영상의 첫째라인 첫째점을 읽을 때 그 영상의 메모리에 쓰고 있는 지점을 체크함.
- ◆ 각 영상의 메모리의 쓰고 있는 지점을 체크하여 한 프레임의 데이터를 읽는 동안 읽고 쓰는 지점이 겹쳐지지 않을 메모리 뱅크에서 읽어나간다.

두 동기가 다른 영상의 최대 시간 차이를 계산해 보면, 한 점의 주기의 최대 차이를 5ns로 보면 최대 시간 차이는

$$\text{Max} = 720 \times 480 \times 5\text{ns} = 1.728 \text{ ms}$$

이 시간은 대략 32라인(Line) 정도의 차이를 가져온다. 따라서 메모리에서 쓰는 지점과 읽는 지점이 32라인 이내로 들어왔을 때 읽었던 프레임을 다시 읽어주면 된다.

4. 실험 및 결과

화면 분할을 하기 위해 구현한 시스템에서 각기 서로 다른 동기를 가진 영상이 선택 알고리즘에 의해 하나의 동기화 제대로 맞추어졌는지 확인하기 위해서 이 데이터를 NTSC 엔코더인 Bt856의 싱크에 동기시켜 읽은 후 모니터로 출력해 보았다.

비교적 화면이 4개의 화상으로 잘 분할되어 출력되고 영상의 픽셀이 깨지는 현상은 발생하지 않았다. 그리고 메모리를 읽고 쓰는 타이밍에 의해서 화면에 줄가는 현상도 나타나지 않았다. 하지만 더욱 선명한 화질을 얻기 위해서는 클럭을 담당하는 부분을 CPLD(ex: Xilinx 7200Series, 카운터 클럭167MHz 까지 지원)등을 사용하

여 지연없이 신호를 출력하도록 해야 한다. 또한, 메모리의 콘트롤 신호가 안정적으로 전달되기 위해서 신호의 맵핑을 막는 직렬 저장(damping register)을 삽입하여 개선해야 한다.

화면 분할하는 시스템을 구현하는데 있어서 전용 칩인 PIP를 사용하지 않고 메모리를 직접 제어하는 로직을 FPGA를 사용해 실험 및 테스트를 실행하여 보았다. 실제 로직을 구현한 FPGA(Xilinx 5200 Series)는 XC5210-5 타입을 사용하였고, 데이터를 저장하기 위한 펌프 메모리로는 최소 저장 시간이 30ns인 μ PD42280-30를 사용하였는데, 화질을 더욱 선명하게 하기 위하여 데이터 저장 시간을 개선해야 한다. 이러한 개선을 위해서 FPGA는 XC5210-4, 프레임 메모리는 μ PD42280-20 등을 사용하여 좀더 빠른 데이터 저장이 가능하도록 하는 것이 바람직 하다.

5. 결론 및 향후 계획

본 연구에서 FPGA(Xilinx 5210-208)를 사용하여 화면 분할 시스템을 구현해 보았다. 동기가 서로 다른 4개 카메라의 동영상을 한 화면에 4분할시켜 디스플레이 하기위해 각 영상마다 2개의 뱅크의 메모리를 두어 1초에 30프레임 정도의 끊어짐 없는 영상을 얻기 위하여 최적의 선택을 하는 선택 알고리즘을 적용해 보았다. 비교적 선명하고 끊어짐 없는 영상을 얻을 수 있었다.

본 연구에서는 4개의 영상만을 가지고 구현을 해보았으나, 더 나아가서 최대 16개의 영상을 1초에 30프레임 정도의 영상을 한 화면에 디스플레이 할 수 있는 시스템을 구현할 계획이다.

< 참고 문헌 >

1. Peter H. Beards, "Analog and Digital Electronics", Prentice Hall, 1991
2. Xilinx, "Development System User Guide", Xilinx Cop., 1992
3. Rogler L. Tokheim, "Theory and Problems of Digital Principles Second Edition", Mc Graw Hill, 1988
4. Xilinx, "The Programmable Logic Data Book", Xilinx Cop., 1996
5. Xilinx, "View Logic Interface Guide", Xilinx Cop., 1995
6. A.Murat Tekalp, "Digital Video Processing", Prentice Hall Signal Processing Series, 1995