칼라 4화면 분할기의 메모리제어 알고리듬에 관한 연구

정회원 손 종 형*, 정 정 화**

A Study on Memory Control Algorithm of a Compact Color QUAD System

Jong-Hyoung Son*, Jong-Wha Chong** *Regular Members*

요 약

본 논문에서는 소형으로 설계된 칼라 4화면 분할기를 위한 메모리 제어 알고리듬을 제안한다. 4화면 분할기는 크게 비디오 디코더부, 메모리부, 비디오 인코더부, OSD (On Screen Display)부, MICOM부, 제어부로 구성되어 있다. 본 논문의 칼라 4화면 분할기는 비디오 디코더부와 비디오 인코더부를 각각 원칩을 이용하여 설계하였으며, 제어부를 FPGA를 사용하여 원칩으로 제작하였다. 화면 4분할을 위해서 메모리 읽기 신호를 실 시간으로 제어하여 비디오 시스템을 제작하였다. 사용된 메모리 제어알고리듬은 비디오신호제어 및 디지털 메모리를 이용하는 다른 시스템에 적용될 수 있다.

Ⅰ. 서 론

사회가 발전함에 따라 안전과 보안에 대한 중요성이 점증하고 있으며, 이에 따라 감시대상지역이 늘어나고 있다. 때문에 수십 대의 카메라를 여러 곳에 설치한 후 한곳에서 이들 카메라로부터 오는 영상을 집중 감시하거나 녹화할 필요가 있다. 집중감시에서는 1개의 화면에 여러 개의 카메라 영상을 동시에 볼 수 있도록 해주는 화면분할기를 사용하여 화면을 4분할[6], 9분할 또는 16분할한 후 집중감시한다. 중요지역을 녹화할 때는 카메라 1대의 영상을 VTR 1대로 녹화하는 것이 아니라 통상 수 내지 수십 대의 카메라 영상을 1대의 VTR에 녹화한다. 이를 가능케 해주는 장비가 프레임스위쳐[7]이다.

지금까지 개발된 화면 분할기[6]는 많은 수의 개별 소자들을 사용하여 설계, 제작되었다. 때문에 제품의 신뢰도가 떨어지고 소비전력이 많으며, 양산비용이 증가하는 단점을 가지고 있다.

한편, VLSI의 발달로 인하여 시스템 온 칩(system on chip)이 진행되어 비디오신호 처리분야에서도 여러 가지 형식의 아날로그 비디오신호를 입력으로 받아 휘도 신호와 색 신호를 분리하고 A/D 변환한 후 디지털로 출력하고 이와 함께 동기신호도 생성하는 칩 즉, 멀티스탠더드 비디오 디코더들을 이용할 수 있게 되었다. 또한, 여러 가지 형식의 디지털비디오 신호와 함께 동기신호를 입력으로 받아 아날로그 비디오 신호로 변환해주는 칩 즉, 비디오 인코더를 이용할 수 있게 되었다. 또, 특정 시스템을 구성하는데 필수적인 컨트롤러의 설계시 종전에는 여러 개의 TTL[9]이나 GAL[1] 등을 이용하여 설계했는데 최근에는 집적도가 높은 FPGA를 이용하여 메모리 컨트롤러를 설계하고 있다.

메모리 제어 기술은 디지털 비디오 처리 시스템은 가장 핵심적인 부분으로, 기존의 일반적인 방법은 2개의 비월 주사시의 2개의 프레임 중에서 하나만을 채택하여 실 시간 display시에 문제를 야기했다. 이러한 문제점을 해결하기 위해서 본 논문에서는 4화면 분할 시스템을 위해서 실 시간 디스플레이 알고리듬을 제안한다. 제안된 방법은 고속 메모

* Gartner Group 한국지사장 겸 반도체 담당 부사장, ** 한양대학교 전자공학과
논문번호 : 99282-0721, 접수일자 : 1999년 7월 21일

리 컨트롤러를 구현하기 위해서 Anti-Fuse방식의 FPGA를 이용해 구현하여 전체 시스템의 구조를 최소화 하고자 한다.[12] 또한, 급격히 발전하고 있는 VLSI 기술을 이용하여 전술한 감시시스템의 문제점을 극복하는 방법을 본 논문에서는 칼라 4화면 분할기를 설계하고 그것을 구현하기 위한 시스템을 제안한다. 본 논문의 칼라 4화면 분할기의 설계에 있어서 비디오 디코더부, 비디오 인코더부 및 컨트롤러부는 각각 원칩으로 구성했다. 또, 분할된 화면상에 채널 명, 시간 등을 표현하기 위하여 OSD 칩을 사용하였으며, PC와의 통신, 각 칩의 제어를 위해 MICOM[5]을 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 아날로그 비디오 신호와 디지털 비디오 신호에 대해서 설명하고 디지털 비디오 신호를 이용하여 화면을 4분할하는 동작원리에 관하여 설명한다. 3장에서는 화면 4분할을 위한 메모리 컨트롤 알고리듬을 소개하고, 4장에서는 칼라 4화면 분할기의 전체 구성을 설명하고, 각 부분에 사용된 칩들을 소개한다. 5장에서는 결론을 맺고 앞으로의 연구방향에 대해서 기술한다.

Ⅱ. 비디오 신호 와 동작

비디오신호의 형태는 크게 주사방식에 따라서 비월주사 방식과 순차주사방식으로 나누고, 일반적인 비디오 시스템은 비월주사 방식을 채택하고 있다. 비월주사 방식은 NTSC, PAL등이 있으며, 우리 나라에서는 NTSC가 사용되고 있다. 본 논문에서는 NTSC 방식에 대응하는 칼라 4화면 분할기에 국한하여 논하므로 NTSC 신호 위주로 설명을 한다.

2.1 아날로그 비디오 신호

2.1.1 NTSC 신호

주로 한국, 북미, 일본, 등에서 사용하고 있는 비디오 신호인 NTSC 비디오신호[10]는 초당 30프레임, 프레임당 2개 필드, 525개 주사선으로 구성되어 있다. 즉, 필드당 262.5개의 주사선이 있다. 프레임의 주기는 1/30초, 필드의 주기는 1/60초이며 1초에는 15750개의 주사선이 있다. 주사선은 화면의 좌측상단에서 우측하단으로 주사하며, 주사가 끝나는 부분에는 동기신호가 들어있다. 동기신호에는 수평방향으로 동기를 맞추는 수평동기신호와 수직방향으로 동기를 맞추는 수직동기신호가 있다. 1프레임의 영상을 표현하기 위해서는 그림 1과 같이 필드1 (홀수필드)을 먼저 주사하고, 필드2 (짝수필드)를 그 다음에 주사한다.[11] 실제로 영상이 들어있는 주사선은 21번 주사선부터 263번 주사선까지와 283번 주사선부터 525번 주사선까지이다.

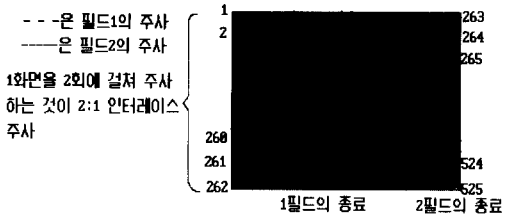

그림 1. NTSC 비디오 신호

2.1.2 PAL 신호

일반적으로 유럽에서 많이 사용되고 있는 비디오 포맷으로 초당 25프레임과 프레임당 2개의 필드 즉 50필드로 구성되어있다. 기본적인 주사선수는 625개로 구성되어 있고, 일반적인 구성은 그림 1과 유사하다.

이러한 비디오 신호를 디지털로 처리하기 위해서 기본적인 디지털 비디오 처리포맷을 ITU-R (CCIR-601)에서 규정하고 있다.

2.2 디지털 비디오 신호 비교

표1 은 NTSC, PAL의 CCIR-601 포맷에 대한 비교표이다. 주사선, 필드 율, 주사선당 픽셀의 수, 프레임당 실제 비디오 신호가 있는 주사선의 개수, 디지털 변환 주파수 등이 나타나 있다.

2.3 NTSC 디지털 비디오 신호

2.3.1 수평주사선

NTSC비디오 신호를 입력으로 받아서 디지털로 신호처리를 행하기 위해서는 아날로그 비디오 신호를 디지털 신호로 바꿀 필요가 있다. 본 논문의 신호처리에서는 IOU-R(CIR-601) 권고 안에서 주어진 YUL 4:2:2 디지털 비디오 신호를 이용한다.[4] 그림 2는 YUV 4:2:2 디지털 비디오신호를 보여준다.

YUV 4:2:2 형식에서는 픽셀 레이트가 13.5MHz이고, 휘도 신호 Y와 색 신호 U,V는 각각 8비트이다.

그림 2. YUV 4:2:2 형식

8비트	$Y_0Y_1Y_2Y_3Y_4Y_5Y_6Y_7$ ..... $Y_{716}Y_{717}Y_{718}Y_{719}$
8비트	$U_0V_0U_2V_2U_4V_4U_6V_6$ ..... $U_{716}V_{716}U_{718}V_{718}$

표 1. 비디오 신호 비교

	NTSC-M	PAL-B/G/H
Line Time(us)	63.55	64.0
Field Rate	60	60
Pixels/Line(N)	858	864
Active Pixels/Line	720	720
Active Lines/Frame	480	580
Pixel Rate(MHz)	13.5	13.5
ADC sampling Rate	3.579	4.43

Y신호는 13.5MHz 속도로 연속해서 픽셀 값이 있으며, U,V 신호는 13.5MHz 속도로 교대로 픽셀 값이 있다. 1개의 주사선상에서 영상신호가 들어있는 부분에 Y신호의 픽셀 수는 720개이고, U, V 신호의 픽셀 수는 각각 360개이다. 그림 3은 CCIR-601포맷에 맞게 나오는 비디오 디코더의 한 주사선 수의 타이밍도를 나타낸 것이다. CK2는 데이터 율, AV는 주사선에서 실제 비디오신호가 있는 신호의 길이를 표시하고, Y,C는 데이터 8비트의 디지털 데이터이다.

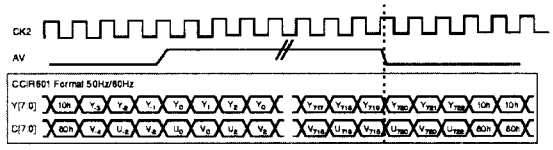

그림 3. 출력 데이터[4]

2.3.2 수직 주사선

그림 4는 NTSC신호의 홀수와 짝수 수직주사선의 타이밍도를 나타낸 그림이다. 그림 4 (a)는 홀수 필드의 그림을 보면 실제 비디오 신호가 생기기전에 동기를 잡기 위한 신호가 있음을 알 수 있다. 각각의 주기는 표1과 동일하다.

그림 4 (b)는 짝수 필드부분을 나타낸 것이다.

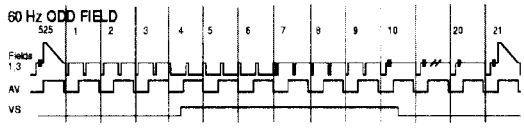

그림 4(a). 홀수 필드[4]

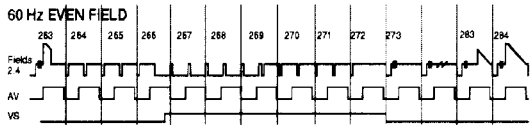

그림 4(a). 짝수 필드[4]

2.4 동작원리

화면의 4분할은 영상데이터를 메모리에 기록하고 읽어내는 도중에 이루어진다. 실 시간에 4개의 영상을 하나의 화면에 표시하기 위해서는 각 영상신호에 대해 수평방향으로 1/2의 영상 데이터를 취하고, 수직방향으로 1/2의 영상데이터를 취한 후, 메모리에 기록한다. 기록된 영상데이터를 읽어낼 때는 각 메모리를 읽어 화면의 4부분을 채운다. 자세한 메모리 제어기 설계 알고리듬은 3장에서 소개한다.

2.4.1 영상 데이터의 기록

수평방향에 대해서 논하자면 그림 2의 YUV 데이터 중에서 그림 5에 주어진 YUV 영상데이터만을 취하여 메모리에 기록한다.

8비트	$Y_0Y_1Y_4Y_5Y_8Y_9$ ..... $Y_{712}Y_{713}Y_{716}Y_{717}$
8비트	$U_0V_0U_4V_4U_8V_8$ ..... $U_{712}V_{712}U_{716}V_{716}$

그림 5. 기록되는 YUV 4:2:2 픽셀

수직방향에 대하여 논하면 그림 1의 주사선 중 21번 주사선부터 263번 주사선까지와 283번 주사선부터 525번까지의 주사선의 데이터 양 중에서 반만 취한다. 주사선 데이터 양의 반을 취하는 방법은 한 개의 주사선씩 건너뛰어서 메모리에 기록한다.

2.4.2 영상 데이터의 읽기

메모리에 기록된 영상데이터를 읽을 때는 분할된 화면의 모양이 그림 4와 같이 되도록 읽어낸다. 메모리를 읽는 클럭은 주파수가 13.5MHz인 클럭을 사용하되 좌측상단을 주사하는 시점에서는 영상1용 메모리, 우측상단을 주사하는 시점에서는 영상2용 메모리, 좌측하단을 주사하는 시점에서는 영상3용 메모리, 우측하단을 주사하는 시점에서는 영상4용 메모리를 읽는다.

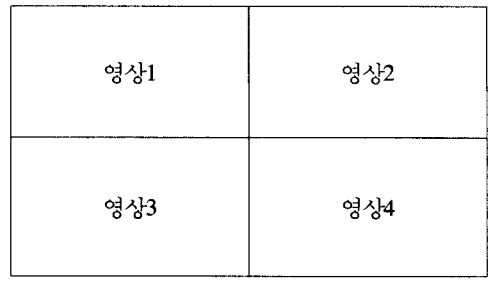

그림 6. 분할된 화면

Ⅲ. 메모리 제어기 설계 알고리듬

본 논문에서는 시스템의 제작을 위해서 FIFO 메모리 제어기를 설계한다. 사용될 메모리는 OKI사의 MSM518221로써 고속의 비디오 신호 처리용으로 적합하다.

3.1 필드 메모리 신호

3.1.1 쓰기 신호

필드 메모리의 쓰기 제어신호는 그림 7과 같이 순차 쓰기 클락(SWCK), 쓰기 리셋(RSTW), 쓰기 가능(WE),입력 가능(IE)신호로 구성되어 있다.

그림 7의 타이밍 신호에 맞게 쓰기 신호를 만들도록 메모리 쓰기 컨트롤을 구성한다.

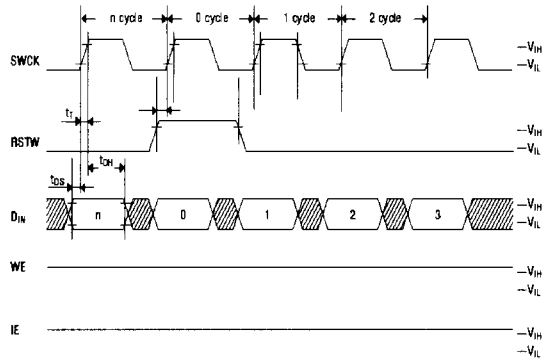

그림 7. 메모리 쓰기 타이밍도

3.1.2 읽기 신호

읽기 신호는 쓰기 신호와 마찬가지로 그림 8의 타이밍도는 순차 읽기 신호(SRCK), 읽기 리셋(RSTR), 읽기 가능(RE), 쓰기 가능(OE) 신호로 구성되어 있다.

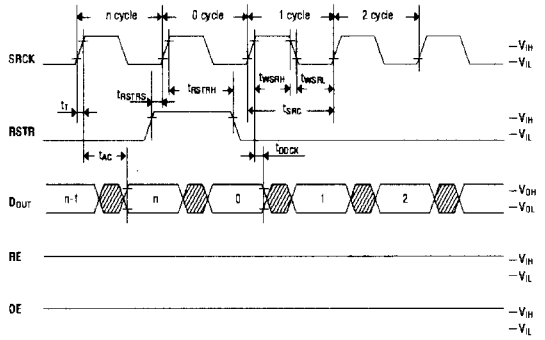

그림 8. 메모리 쓰기 타이밍도

3.2 메모리 제어신호의 생성

메모리 읽기 신호와 쓰기 신호가 어떻게 구성되는지 이 절에서 설명을 한다.

3.2.1 메모리 쓰기 신호

3.2.1.1 기존의 메모리 쓰기 신호

일반적인 4화면 분할기의 메모리 쓰기 신호는 비월 주사방식의 특징을 이용한다. 비월주사 방식의 비디오신호는 한 개의 프레임이 짝수 및 홀수 필드로 구성되어 있는데 화면 4분할을 위해서는 둘 중의 하나의 필드를 가지고 메모리 제어 신호를 쓰기 된다. 그럴 경우 두 개의 필드가 가지고 있는 화면의 실 시간성이 손상되어 화면 표시 시에 느리게 나타난다. 이런 문제점을 해결하기 위해서 두 개의 필드 중에 하나를 사용하는 것이 아니라 두 개의 필드 모두를 사용하여 메모리 제어를 하는 알고리듬이 필요하게 된다.

3.2.1.2 본 논문의 메모리 쓰기 신호

4화면 분할을 위해서 각각의 화면을 각각의 데이터 저장 메모리에 써주기 위해서 그림 10 같은 VHDL 코드로 메모리 쓰기 신호를 만들어 내었다. 4개의 채널이므로 같은 쓰기 모듈이 4개가 필요하다. 4개의 입력신호와 4개의 출력신호를 가지고 메모리 쓰기를 제어하게 된다.

각 채널당 메모리 쓰기 신호도는 그림 9에서와 같이 주어져있다. WE(write enable)신호는 두 개의 픽셀을 건너뛰어 메모리에 써주게 되어있고, 쓰기 클럭(WCLK: write clock)은 13.5MHz의 주기로 데이터를 써주게 된다. 그림 10 은 한 개의 채널의 메모리 맵의 모양을 나타낸 것이다. 하나의 픽셀이 8비트씩이고 Y,C신호각각을 위한 메모리 맵으로써 하나의 칸은 주사선 하나를 나타내고 있고 위에 있는 번호는 픽셀의 개수를 나타내고 있다. 한 주사선당 360개의 픽셀을 메모리에 쓰고, 그러한 주사선 240개를 쓰고 있음을 보여 주고 있다. 이러한 메모리 할당을 각 채널당 하나씩 4개를 이용해서 메모리에 기록하게 된다.

3.2.2 메모리 읽기 신호

화면 분할을 위한 핵심적인 부분으로써 그림11과 같은 메모리 맵을 가진 화면 표시를 위한 제어신호를 만들어 내게 된다. 4개의 화면을 하나의 화면으로 만들어 내려면 각각의 화면데이터 양이 전체데이터 량의 1/4를 앞 절에서 나타내었다. 그림 11을

보면 각 메모리 블록에 쓴 데이터를 한 화면에 나타내기 위해서 실 시간으로 메모리를 읽어 들였다. 즉 한 개의 주사선에서 중간지점(361번째 픽셀) 에서 다른 채널로 넘어가는 스위칭 회로를 만들어 내고 수직 주사선 중에서 중간의 주사선(241번째 주사선)일 때 다른 채널로 넘어가는 메모리 제어 신호를 만들어 내어서 화면 분할을 이루게 된다. 본 논문에서는 일반적으로 다른 화면 분할시스템에서 채택하는 짝수나 홀수 필드 중에서 하나를 버리는 시스템을 채택하지 않고, 모든 필드를 다 사용하는 메모리 제어 시스템을 설계했다.

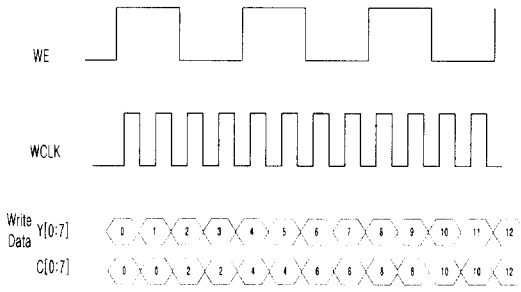

그림 9. 쓰기 타이밍도

	1 2 3 4 ... 357 358 359 360
1	Y0 Y1 Y4 Y5 ... Y712 Y713 Y716 Y717
2	Y0 Y1 Y4 Y5 ... Y712 Y713 Y716 Y717
3	Y0 Y1 Y4 Y5 ... Y712 Y713 Y716 Y717
4	Y0 Y1 Y4 Y5 ... Y712 Y713 Y716 Y717
237	Y0 Y1 Y4 Y5 ... Y712 Y713 Y716 Y717
238	Y0 Y1 Y4 Y5 ... Y712 Y713 Y716 Y717
239	Y0 Y1 Y4 Y5 ... Y712 Y713 Y716 Y717
240	Y0 Y1 Y4 Y5 ... Y712 Y713 Y716 Y717

	1 2 3 4 ... 357 358 359 360
1	U0 V0 U4 V4 ... U712 V712 U716V716
2	U0 V0 U4 V4 ... U712 V712 U716V716
3	U0 V0 U4 V4 ... U712 V712 U716V716
4	U0 V0 U4 V4 ... U712 V712 U716V716
237	U0 V0 U4 V4 ... U712 V712 U716V716
238	U0 V0 U4 V4 ... U712 V712 U716V716
239	U0 V0 U4 V4 ... U712 V712 U716V716
240	U0 V0 U4 V4 ... U712 V712 U716V716

그림 10. 쓰기 데이터의 메모리 맵

3.3 전체 메모리제어 상태도

그림 12는 쓰기와 읽기를 복합한 메모리 제어 상태도를 나타낸 것이다. FIFO메모리 제어이기 때문에 다른 입력과 출력을 독립적으로 억세스 할 수가 있다. 먼저 전체 메모리를 리셋(Memory Reset)한 다음 메모리에 쓰기 가능(WE : Write Enable), 쓰기 클럭(WCLK)신호를 전술한 내용과 같이 만들고, 데이터를 읽을 때는 RE, RCLK를 거치고 주기적으로 메모리를 계속해서 리셋 해준다. 메모리의 뱅크 어드레스는 내부적으로 자동적으로 증가하게 된다.

```
library ieee;
        use ieee.std_logic_1164.all;
        use ieee.std_logic_unsigned.all;

entity exam is
        port (  avi, vsi, frz_i, ck2i,oddi : in std_logic;
        we, ic, rstw, swck          : out std_logic);
end;
architecture exam of exam is
signal odd_pre          : std_logic ;
signal av_inv : std_logic ;
signal av_buf           : std_logic ;
signal vs_buf : std_logic ;
signal vs_buf1          : std_logic ;
signal vs15             : std_logic ;
signal ck2_buf          : std_logic ;
signal frz15            : std_logic ;
signal q                : std_logic ;

signal ck2_buf_cnt      : std_logic_vector(2 downto 0);
signal v_enable         : std_logic ;

begin
        av_inv <= (NOT avi);
        av_buf <= avi;
        odd_pre <= oddi;
        process(vs15)
        begin
                if vs15'event and vs15 = '1' then
                        frz15 <= frz_i;
                end if;
        end process;
        process(vs_buf)
        begin
                if vs_buf'event and vs_buf = '1' then
                        q <= odd_pre;
                end if;
        end process;
        vs15 <= vs_buf AND (NOT q);
        vs_buf <= vsi;
        vs_buf1 <= vsi;
        rstw <= vs15 AND (NOT av_buf);
        ck2_buf <= ck2i;
        swck <= ck2_buf;
        process(ck2_buf)
        begin
                if ck2_buf'event and ck2_buf = '1' then
                ck2_buf_cnt <= ck2_buf_cnt + "001";
                end if;
        end process;

        process(ck2_buf)
        begin
                if ck2_buf'event and ck2_buf = '1' then
                we <= v_enable AND ck2_buf_cnt(1) AND
                ck2_buf_cnt(0) AND (NOT frz15);
                ie <= v_enable AND ck2_buf_cnt(1) AND
                ck2_buf_cnt(0) AND (NOT frz15);
                end if;
        end process;
end exam;
```

그림 11. 쓰기 VHDL 코드

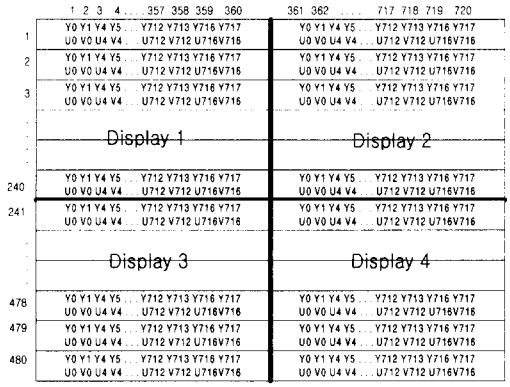

그림 12. 읽기 데이터 맵

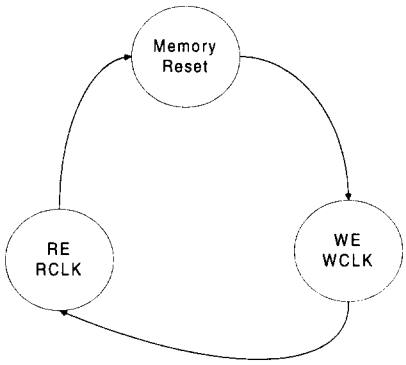

그림 13. 메모리 제어 상태도

3.4 화면 디스플레이 전환

그림 11에서처럼 다른 화면을 스위칭 해야만 화면을 나눌 수가 있는데 그림 13은 수직동기신호와 수평동기 신호를 이용해서 메모리 뱅크스위칭 하는 그림을 나타내었다. 수직 및 수평 동기신호를 중간 지점에서 세어서 그 순간에 화면이 전환되도록 설계를 해야 한다. 2/4디코더를 이용하여 각각의 화면으로 스위칭 되도록 한다.

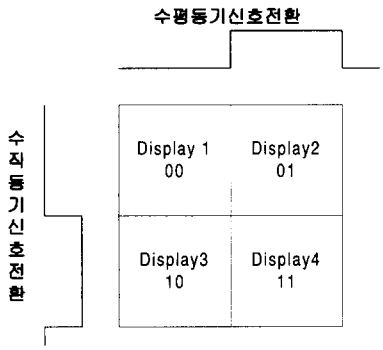

그림 14. 메모리 맵의 전환

Ⅳ. 설 계

3장에서 기술한 동작원리에 따라서 동작하는 소형의 칼라 화면 4분할기를 만들기 위해서는 비디오 디코더부, 비디오 인코더부, 메모리부, 제어부, 마이콤부, OSD부 등을 설계할 필요가 있다. 본 논문에서 설계한 회로의 전체 구성은 그림 15와 같다.

화면 4분할기에서 처리해야할 영상신호의 수가 4개이기 때문에 비디오 디코더부의 크기를 우선적으로 줄이고, 그 외의 크기를 줄이는 것이 바람직하다. 그래서, 현재 시중에서 이용 가능한 부품들 중에서 각 부를 원 칩으로 해결할 방법을 찾았다.

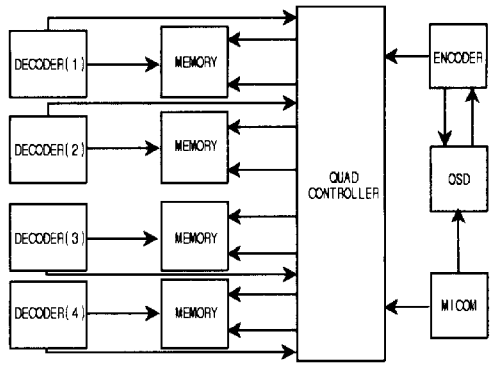

그림 15. 4분할기 구성

4.1 비디오 디코더부

비디오 디코더부는 삼성의 비디오 디코더 KS0122[4]를 사용하여 구성했다. 이 칩은 NTSC, PAL, SECAM 뿐 아니라 S 비디오에도 대응할 수 있도록 설계되어 있고, 디지털 출력의 형식도 다양하다. 본 논문의 4분할기에서는 이 칩을 마이콤으로 제어할 때 아날로그 NTSC 신호로부터 색 분리를 행한 후 A/D 변환하고 YUV 4:2:2의 디지털 비디오 데이터를 13.5MHz로 출력하도록 했다. YUV 4:2:2의 데이타는 그림 2와 같이 16비트로 출력된다. 이 칩은 또한 비디오 데이터와 함께 필드판별신호, 수평동기신호, 수직동기신호, 13.5MHz 클럭 등을 출력한다.

4.2 비디오 인코더부

비디오 인코더부는 삼성의 비디오 인코더 KS0119Q2[4]를 사용하여 구성했다. YUV 4:2:2의 디지털 비디오 데이터를 13.5MHz로 입력으로 받아 NTSC 아날로그 비디오 신호를 출력하도록 이 칩을

프로그램 했다. 또, 이 칩이 자체적으로 비디오 신호 관련신호 - 필드판별신호, 수평동기신호, 수직동기신호, 13.5MHz 클럭 등을 출력하도록 마스터 모드에서 동작시켰다.

4.3 메모리부

메모리부는 컨트롤러의 크기를 줄이기 위하여 OKI사의 FIFO인 MSM518221[2]을 사용했다. 입력 핀과 출력 핀이 분리되어 있는 이 칩의 특성을 이용하여 완전히 비동기로 동작하는 카메라간의 동기를 맞추기 쉽게 하고 컨트롤러의 크기를 줄이려고 했다.

4.4 제어부

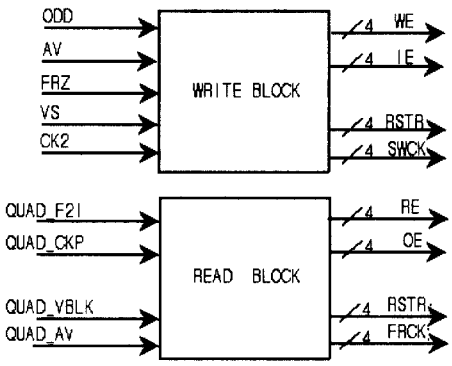

그림 16. 제어부의 구성

제어부는 Quick Logic사의 FPGA QL12*16B -184PLC[8]를 이용하여 원 칩으로 설계했다. 제어부에서는 3장에서 기술한 동작원리에 따라 영상 데이터를 메모리에 기록하고 읽어내며, 비디오 디코더부와 비디오 인코더부의 동기신호에 대응하여 기록과 읽기를 행한다. 또한, 마이콤의 제어신호에 따라 화면정지 기능을 제공한다. 제어부의 회로구성은 그림 15와 같다.

컨트롤러는 디지털 영상을 메모리에 기록하는 블록과 메모리로부터 디지털 영상을 읽는 블록으로 구성되어 있다.

4.5 마이콤부

마이콤부는 삼성의 KS88P0016[5]를 이용하여 구성했다. 이 마이콤부는 비디오 디코더, 비디오 인코더, 제어부, OSD등을 초기화하고 PC와의 AS232C 통신을 수행한다. 따라서, 원격에서 본 논문의 칼라 4화면 분할기를 제어할 수 있도록 한다.

4.6 OSD부

OSD부는 삼성의 KS5514-2[3]을 이용하여 구성하였으며, 아날로그 영상신호를 입력으로 받아서 글자가 삽입된 아날로그 영상신호를 출력한다. OSD부에서는 마이콤의 명령에 따라 년, 월, 일, 시, 분, 초, 요일, 채널 명, 화면정지여부를 표시한다.

4.7 실험 및 PCB제작

4.7.1 실험 및 측정

본 논문의 칼라 화면 4분할기는 16비트 YUV 4:2:2 형식으로 신호처리를 행하고 있어서 화질이 뛰어나고, 초당 30프레임을 처리하고 있으므로 움직이는 물체의 움직임이 원래의 화상과 같이 자연스럽다. 또, 신호처리의 주요 부분을 원 칩으로 설계하였으므로 보드의 크기가 소형화되었고 전력소비 또한 적은 장점을 가지고 있다. 그림 17과 그림 18의 파형은 쓰기 및 읽기 파형의 측정 결과이다. 그림 17의 위쪽 파형은 쓰기 제어 신호를 나타낸 것이고, 아래쪽 파형은 쓰기 데이터 픽셀 율 13.5MHz의 파형을 표시한 것이다. 그림 17의 위쪽

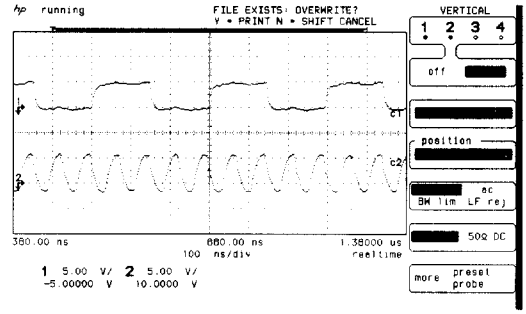

그림 17. 쓰기 신호의 측정 파형

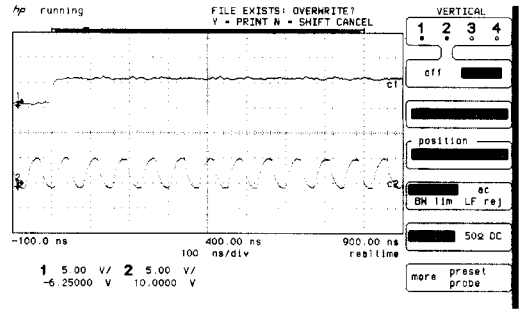

그림 18. 읽기 신호의 측정 파형

파형을 분석해보면 두 개의 픽셀을 건너뛰고 쓰기 신호를 만들어 내는 것을 알 수가 있다.

4.7.1.1 결과에 대한 고찰

3장의 그림 9에서처럼 메모리에 데이터를 써주기 위한 파형을 메모리 컨트롤러를 이용하여 그림 17의 파형으로 생성된다. 메모리 컨트롤러의 파형이 실제 예상한 파형과 일치하므로 정확히 데이터가 메모리에 써짐을 알 수가 있다.

4.7.2 제작

설계된 칼라 4화면 분할기의 성능을 평가하기 위하여 4층 PCB로 제작하고 실험하였다. 그림 7은 4화면 분할기 PCB의 사진이며, 그림 8은 제작된 4분할기에 카메라를 연결하여 화면을 출력한 것이다.

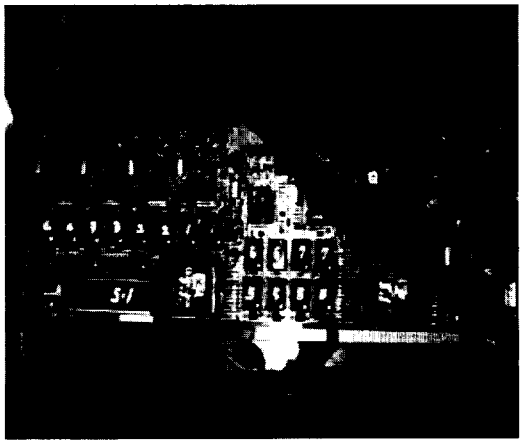

그림 19. 4화면 분할기의 PCB

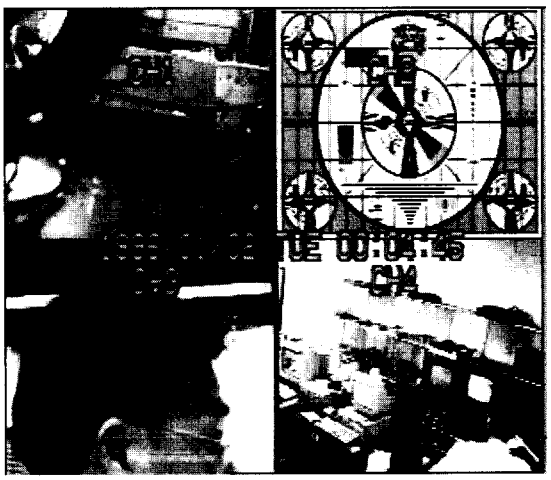

그림 20. 4분할 화면

V. 결 론

본 논문에서는 칼라화면 4분할기를 설계하고, PCB로 제작하였다. 특히, 비디오 신호처리의 주요 부를 원칩을 이용하여 PCB의 크기를 최소화하였다. 화상 데이터를 읽기 위한 메모리 제어 신호를 이용하여 메모리에 데이터 쓰기 신호와 읽기 신호를 생성하였다. 화면을 실 시간으로 표현하기 위해서 메모리 쓰기 신호를 생성할 때 양쪽 필드를 모두 사용했고, 각각의 채널이 비동기 적으로 동작할 수 있도록 설계했다. 또한 본 연구에서 만들어진 시스템은 디지털 비디오 시스템 구조에 응용될 수 있다.

참 고 문 헌

[1] LATTICE, pLSI and ispLSI Data Book and Handbook, 1992.

[2] OKI, MEMORY LSI DATA BOOK, 1994.

[3] SAMSUNG, VIDEO IC DATA BOOK, 1995.

[4] SAMSUNG, Multi Media DATA BOOK, 1996

[5] SAMSUNG, KS88C0016 Microcontroller User's Manual, 1993.

[6] SAMSUNG, 4분할기 매뉴얼, 1996.

[7] SAMSUNG, 프레임 스위처 매뉴얼, 1996.

[8] QUICK LOGIC, QUICK LOGIC DATA BOOK, 1995.

[9] LS-TTL IC DATA BOOK, 세화출판사, 1992.

[10] 기초부터의 영상신호처리, 세화출판사, 1988.

[11] 첨단, 월간 전자기술, 1995. 9.

[12] 이영호, 김민호, 한기웅, 정정화, "칼라 4화면 분할기의 소형화에 관한 연구", CAD 및 VLSI 설계연구회 학술발표회 논문집, 대한전자공학회, pp.120-124, 1996

손 종 형(Jong-Hyoung Son) 정회원

Vol 24. No. 10A호 참조

현재 : Garthner Group 한국지사장 겸 반도체 부사장

정 정 화(Jong-Wha Chong) 정회원

Vol 24. No. 10A호 참조

현재 : 한양대학교 전자공학과