

論文2003-40SC-6-10

CPLD를 이용한 Monochrome/Color 실시간 변환기 설계 및 구현

(Design and Implementation of CPLD-Based Monochrome to Color Real Time Converter)

尹宰茂*, 姜雄紀*, 陳台錫*, 李章明*

(Jae-Mu Yun, Woong-Ki Kang, Tae-Seok Jin, and Jang-Myung Lee)

요약

텍스트 모드에서 흑백 데이터를 컬러 데이터로 변환시킬 때 하드웨어적인 방법을 이용하여 실시간적으로 변환할 수 있는 회로를 설계 및 구현하였다. Color Palette ROM에 화면을 구성하는 모든 페이지에 해당되는 색상 정보를 저장한다. 이 색상 정보는 8비트 단위로 출력이 되며 하위 4비트는 전경색, 상위 4비트는 배경색을 가지도록 지정한다. Address Reduction ROM을 두어 중복되는 어드레스를 배제하여 Color Palette ROM의 용량을 1/16로 감소시킬 수 있도록 하였으며, 다수 개의 D-FF을 두어 어드레스와 데이터 그리고 페이지 정보를 임시로 저장한 후에, Counter를 통해 실시간에 8회의 처리과정을 거친 후 Multiplex에서 전경색과 배경색을 구분하여 컬러 비디오 컨트롤러에 색상 정보를 보내도록 설계하였다. 기존의 흑백 LCD 디스플레이를 사용하고 있는 각종 제어장치 설비를 컬러로 변환함에 있어서 용이한 실시간 인터페이스로 활용될 것이다.

Abstract

When we transformed from Monochrome-data to Color-data in text mode, we used hardware-method to design the circuit which is convertible in real time. We saved color information in every screens that can make screen in Color Palette ROM and it is also generated 8bit. lower 4bit assign foreground color and upper 4bit can design to have background color. We have Address Reduction ROM to remove repeated address and reduce volume of Color Palette ROM to 1/16. Besides, we have many D-FF to save address, data and page information temporarily after that, we have management process 8 times through counter in real time. Finally, we chose either foreground color or background color in multiplex and established color information was sent to the color video controller. Thus, you can use it as a good interface when you transfer many control devices with Monochrome display(ex, LCD Monitor) into devices with Color display.

Keyword : Color Palette ROM, Address Reduction ROM, D-FF, Counter, Multiplex.

I. 서론

최근 많은 산업용 기기들의 화면표시 장치들이 흑백에서 컬러화 되어가고 있는 추세이다. 심지어 공학용 계산기나 휴대폰과 같은 소형 기기에까지도 컬러화면 표시장치를 도입하게 되었다. 과거에 이미 개발하였던 흑백 기반의 장치들을 컬러화 하는 데에는 비디오 장치를 교체하여야 하고 진보된 많은 프로그래밍 작업이

* 正會員, 釜山大學校 電子工學科 지능로봇실실험실
(Department of Electronics Engineering Pusan National University)

接受日字:2003年6月10日, 수정완료일:2003年9月8日

필요하다. 색상정보를 고려한 컬러 작업의 프로그래밍을 하다보면 소스코드가 증가하게 되고 전체적인 시스템의 속도저하 문제가 발생한다. 그리고 흑백에서처럼 빠른 출력의 화면 스크롤은 기대를 할 수 없게 된다. 전체적인 시스템을 Upgrade 해야만 속도 문제를 극복할 수 있는데, 시스템을 Upgrade 함에 따라 가격이 상승하고 개발 기간이 길어지는 문제점이 발생하게 된다. 이에 대비하여 기존의 흑백 장치에서 이미 개발되었던 프로그램을 거의 수정하지 않고도 간단한 결합 작업만으로 빠른 속도를 유지한 채 컬러화 할 수 있는 기기의 개발이 요구된다.

본 논문에서는 기존의 흑백 표시장치의 정보를 그대로 둔 채 컬러화 하도록 설계하는 것이며, 어느 시스템에서나 쉽게 응용이 가능하도록 구현하였다. 또한 Color palette ROM을 두어서 사용자가 원할 때 언제든지 화면색상을 쉽게 변경할 수 있도록 설계하였다. 모든 동작은 하드웨어적으로 처리되므로 입/출력에 따른 지연현상이 거의 없고, ROM과 CPLD(Complex Programmable Logic Device)의 시간 지연만이 고속처리 제한으로 작용한다. 현재 구현된 것은 텍스트 모드, 640×400 해상도, 65535 컬러 중에서 16 컬러를 사용하는 것으로 제한을 두었지만 응용에 따라서 더 높은 해상도와 많은 색상을 사용할 수 있다.

아무리 속도가 빠르고 편리할 지라도 산업계에서는 가격 상승으로 인해 기기가 보급되지 않는 경우가 많다. 이런 측면에서 색상정보가 내장된 Color palette ROM의 용량을 최소화하기 위하여 Address reduction ROM을 두어서 텍스트 화면의 색상 중복 여부를 분석하여 최소한의 색상정보로써 컬러화 할 수 있는 알고리즘을 만들었다. 알고리즘의 결과 값을 바탕으로 ROM에 기입을 하고 Color palette ROM의 주소와 결합을 하면 Color palette ROM의 용량을 대폭 축소시킬 수 있는 장점을 가지고 있다.

II. 흑백과 컬러 화면의 구성

640×400의 해상도를 가지는 흑백과 컬러 화면에서 어드레스 증가에 따라서 어떻게 화면이 표시되는지 그림으로써 나타낼 수 있다. 제품마다 서로 다른 특성이 있으나 아래에 나열된 것들은 가장 일반적인 화면의 1를 든 것이다.

<그림 1>의 흑백 화면의 구성을 살펴보면 8개의 픽

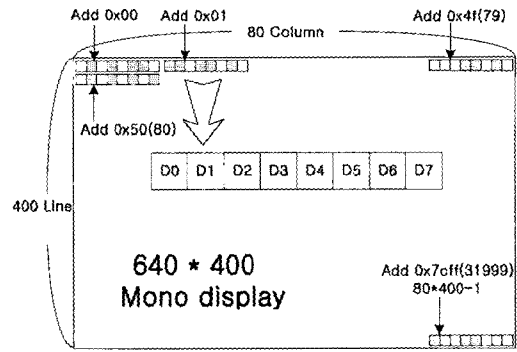


그림 1. 흑백 화면의 구성
Fig. 1. Composition of monochrome screen.

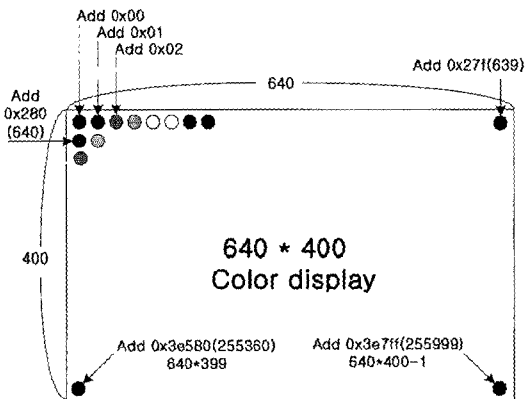


그림 2. 컬러 화면의 구성
Fig. 2. Composition of color screen.

셀(Pixel) 단위로 가로 80 Column, 세로 400 Line으로 구성되어 있다. 어드레스가 한 번 증가할 때마다 8개의 픽셀이 화면에 나타나게 된다. 한 화면을 모두 나타내기 위해서는 32000번(80 Column × 400 Line) 어드레스가 증가되어야 한다. 8개의 픽셀을 분석해보면, 비트의 배열순서는 D7~D0의 순서가 아닌 D0~D7의 순서로 구성되어 있다.

<그림 2>의 컬러 화면의 구성에서는 어드레스가 한 번 증가할 때마다 하나의 픽셀이 화면에 나타나고 정의된 색상을 띄고 있다. 어드레스가 한 번 증가할 때마다 하나의 픽셀 단위로 화면을 구성하므로 화면을 모두 나타내기 위해서는 256000번(640 Pixel × 400 Line) 어드레스가 카운트가 될 필요성이 있다.

이것으로 컬러 화면이 흑백 화면보다 8배에 해당하는 3개의 어드레스 라인이 더 필요하다는 것을 알 수 있다.

<그림 3>의 흑백 제어기의 구조를 살펴보면 어드레

스 라인이 15개, 데이터 라인이 8개로 이루어져 있다. 15개의 어드레스로 제어할 수 있는 영역은 $2^{15} = 32768$ 번지까지 제어가 가능하다. 흑백 화면에서는 한번에 8개의 픽셀 단위로 화면이 구성되므로 640×400 해상도에서는 $32000(640/8 \times 400)$ 에 해당하는 어드레스의 증가가 필요하므로 15개의 어드레스 라인으로 구성할 수 있다.

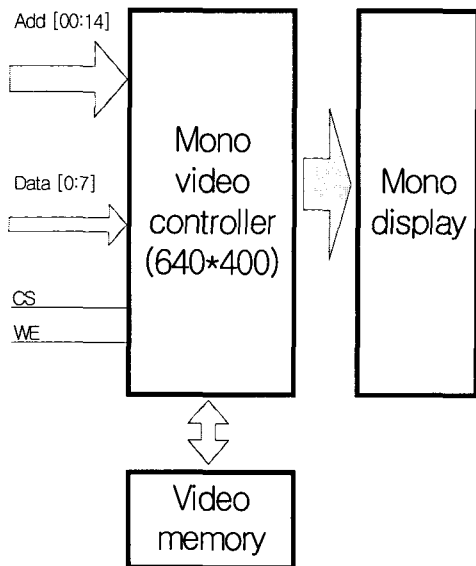


그림 3. 흑백 제어기의 구조
Fig. 3. Structure of mono controller.

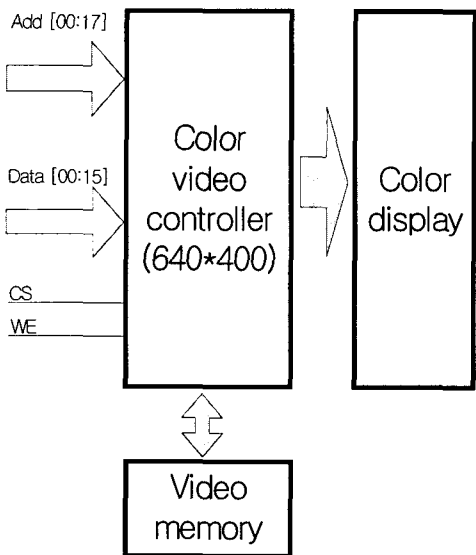


그림 4. 컬러 제어기의 구조
Fig. 4. Structure of color controller.

표 1. 흑백과 컬러 제어기의 비교

Table. 1. Comparison between monochrome and color controller.

구분	Monochrome controller	Color controller
Data bus	8	16
Address bus (640*400 Mode)	15	18
8 Pixel display	1 Address count	8 Address count

<그림 4>의 컬러 제어기의 구조를 살펴보면 어드레스 라인이 18개, 데이터 라인이 16개로 이루어져 있다. <그림 2>에서와 같이 컬러 화면은 하나의 어드레스가 증가될 때마다 하나의 픽셀만을 나타낼 수 있으므로 총 어드레스 라인은 18개($2^{18} = 262144$, $640 \times 400 = 256000$)가 필요하다. 그리고 데이터 버스 16개는 하나의 픽셀에 해당하는 색상인 65535 컬러($2^{16} = 65536$)를 나타낸다. 본 논문에서는 16 컬러를 사용하므로 65535 컬러 중에서 사용자가 원하는 색상 16개를 임의로 선택할 수 있다.

이것으로 640×400 모드에서 흑백 제어기와 컬러 제어기의 차이점은 <표 1>과 같이 정리할 수 있다. 가장 큰 차이점 두 가지를 든다면 흑백제어기는 하나의 어드레스가 증가될 때마다 8개의 픽셀이 화면에 나타나고, 컬러 제어기에서는 하나의 어드레스가 증가될 때마다 하나의 픽셀이 화면에 나타난다. 그리고 각 픽셀에 65535 컬러를 나타낼 수 있다. <표 1>을 바탕으로 CPLD(Complex Programmable Logic Device)를 이용하여 하드웨어적으로 흑백을 컬러 색상으로 실시간 변환시킬 수 있다.

III. 중복된 색상정보의 축소법

<그림 5>는 'A', 'B', 'C'라는 글자를 Color palette ROM을 통과한 후에 컬러로 변환되는 글자들을 보여준다. Color palette ROM에서 하나의 문자에 해당되는 색상정보는 가로 8비트 단위로 세로 16번만큼의 영역을 확보하고 있다. 그림에서와 같이 16개의 8비트에서 하위 4비트는 전경색(Background color)을 상위 4비트는 배경색(Foreground color)을 지정해 준다. 본 제안에서는 4비트를 사용하여 나타낼 수 있는 색상을 16가지

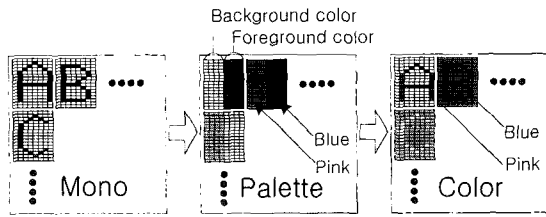


그림 5. 1 : 1 컬러 변환
Fig. 5. Direct conversion from Monochrome to Color.

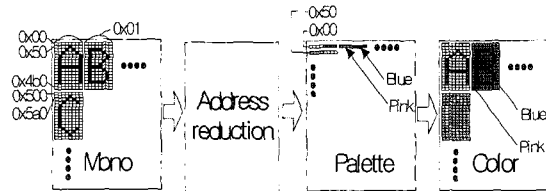


그림 6. 제안된 방법.
Fig. 6. Proposed method.

로 제한하였다. 현재는 Color palette ROM을 색상으로 나타내었으나 실제로는 숫자정보가 ROM에 8비트 단위로 기입되어 있는 형태로 구성된다. 컬러 변환의 예로 'B'의 글자를 흑백에서 컬러로 변환시키기 위해서는 Color palette를 가로로 8번 세로로 16번 거치게 된다. 문자 'B'에 해당되는 Color palette ROM을 보면 전경색은 Blue를 배경색은 Pink의 색상으로 지정되어 있음을 알 수 있다. 흑백의 데이터는 Color palette ROM을 거치면서 각각의 비트가 'H'와 'L'에 따라서 전경색과 배경색 중에서 한 가지 색상을 컬러로 나타낸다. <그림 5>에서는 모든 흑백정보의 크기와 동일하게 Color palette ROM을 구성해야 하므로 실무에서 사용할 때는 용량이 커지게 되어 Cost가 증가하는 요인이 된다.

본 연구에서 제안한 방법으로 구성한 것이 <그림 6>이다. Address reduction ROM을 하나 추가함으로써 Color palette ROM의 용량을 대폭 줄일 수 있는 방법을 보여준다. 예로 'A'라는 문자를 흑백에서 컬러로 변환시킬 때 Color palette ROM에서는 8비트 데이터 16개를 모두 사용하지 않고 한 개를 사용하여 컬러로 변환시킬 수 있다. 이렇게 한 개로써 가능한 이유는 한자의 문자를 나타내기 위하여 일반적으로 전경색과 배경색을 한가지씩만 사용하므로 <그림 5>와 같이 8비트 단위로 세로로 16번의 색상 정보가 모두 동일하기 때문이다. 즉, 0x00, 0x50, 0xa0, 0xf0, 0x140, 0x190, 0x1e0, 0x230, 0x280, 0x2d0, 0x320, 0x370, 0x3c0,

0x410, 0x460, 0x4b0 번지에 해당되는 팔레트는 0x00 번지의 팔레트 정보와 동일하며 16번 반복이 된 것이다. 이것을 고려하여 Address reduction ROM을 중간에 두어서 색상이 중복되는 어드레스 16개를 하나의 어드레스로 바꾸어주는 역할을 함으로써 Color palette ROM의 용량을 1/16로 감소시킬 수 있다.

<그림 7>은 텍스트 모드에서의 문자 "A"의 구성을 보여준다. 가로로 80자, 세로로 25줄을 사용하며, 한 문자를 위해 가로로 8개, 세로로 16개의 픽셀을 사용하는 것을 보여준다.

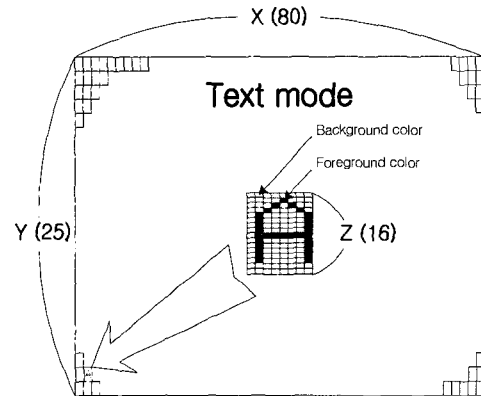


그림 7. 텍스트 모드
Fig. 7. Text mode.

```

for k=0:Y-1
{
    for j=0:Z-1
    {
        for i=0:X-1
        {
            out(X*k+i)
        }
    }
}
    
```

그림 8. 어드레스 축소를 위한 알고리즘
Fig. 8. Algorithm for address reduction.

<그림 8>은 Address reduction ROM을 사용하여 Color palette ROM의 용량을 1/16로 축소하기 위한 알고리즘을 보여준다. <그림 7>을 참고하여 X와 Y는 텍스트 모드에서 가로와 세로의 글자 수이고, Z는 하나의 글자를 위한 세로 픽셀 개수를 나타낸다. 실제 알고리즘에서는 X=80, Y=25, 및 Z=16을 미리 대입하였다.

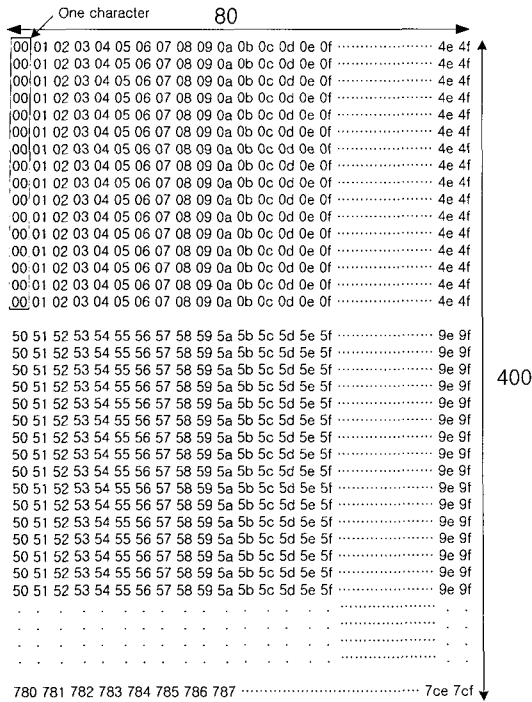


그림 9. 수식 적용 결과
Fig. 9. The result of the applied formula.

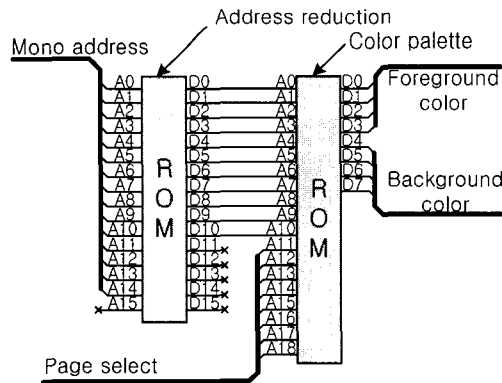


그림 10. Address reduction ROM과 Color palette ROM의 실제 구성
Fig. 10. Real composition of Address reduction ROM and Color palette ROM.

<그림 9>는 <그림 8>의 알고리즘에 의한 값을 구한 후 80개 단위로 자르고 다시 세로로 16개씩 분류를 하여 나타낸 것으로, 결과 값을 ROM에 기록하면 어드레스 변환기를 만들 수 있다. 그림의 네모 박스내의 데이터와같이 어드레스가 80(0x50) 단위로 0x00, 0x50, 0xa0, 0xf0, 0x140, 0x190, 0x1e0, 0x230, 0x280, 0x2d0, 0x320, 0x370, 0x3c0, 0x410, 0x460, 0x4b0 중에서 임의

의 번지가 지정되면 출력은 모두 동일하게 0x00이 되게 하는 것이다. 이것은 <그림 7>에서 (X, Y, Z)에 대응하는 주소를 한 개의 새로운 주소 (X, Y)로 바꾸어주는 것으로 전체의 ROM의 크기를 1/16로 줄이는 기법이다. 즉, Z = 0~15까지 변화하는 동안 (X, Y, Z) 좌표를 모두 동일한 (X, Y)로 맵핑하게 만드는 것이다.

<그림 10>의 Address reduction ROM에는 <그림 9>와 같이 나열된 데이터 값이 기록되어 있다. 즉, 흑백 어드레스가 입력되면 기록된 데이터 값이 출력되는데, 데이터 값 자체가 중복을 배제한 축소된 어드레스가 되어 다시 Color palette ROM의 어드레스로 활용된다. 그리고 Color palette ROM의 어드레스의 상위 부분은 Page select의 목적으로 활용된다. 즉, <그림 11>과 같이 8개의 Page select 비트들에 의하여 총 256 페이지까지의 서로 다른 화면의 색상을 구성할 수 있다.

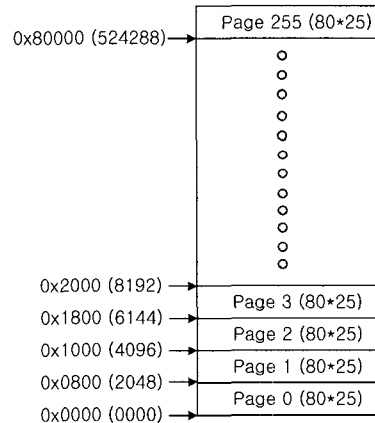


그림 11. Color palette ROM의 구성
Fig. 11. Composition of Color palette ROM.

비교적 응답 속도가 느린 ROM을 두 번 거치게 되면 시간 지연이 많이 발생하여 실시간 처리에 어려움이 있을 것이라는 예상을 할 수도 있다. 하지만 최근 ROM의 동향 및 추세를 볼 때 과거에 120ns의 속도 지연에서 35ns 정도로 속도가 대폭 빨라지고 있으며, 또한 NV RAM(Non Volatile Read Only Memory)을 사용한다면 15ns 정도로 액세스 시간을 대폭 줄일 수 있다^[2]. 현재 두 개의 ROM을 직렬로 연결하여 사용하는 경우는 시간 지연이 대략 70ns 정도이다. 640×400 해상도를 갖는 표시장치에서 시간 지연을 계산하면, 흑백 데이터는 8개의 픽셀 단위로 기록이 되므로 한 화

면을 구성하기 위해서는 80×400번 어드레스 변경이 필요하다. 따라서 ROM을 지나는 총 지연 시간은 한 픽셀의 표시를 위하여 ROM을 2회 사용함에 의한 시간지연 70ms를 고려할 때 이론적으로 한 화면을 띄우기 위한 시간은 $80 \times 400 \times 70^{-9} = 2.24ms$ 라는 시간이 걸린다. 이것은 시각적으로는 거의 무시해도 좋을 만큼 짧은 시간임을 알 수 있다.

IV. 전체 시스템의 구성

<그림 12>의 전체 구성도에서 좌측 부분의 Mono data, Mono address, Mono WR 신호는 기존의 흑백 제어기에서 사용하는 신호를 나타낸다. CPLD를 기준으로 우측 부분은 흑백 데이터가 컬러 데이터로 변환이 되어 컬러 제어기에 전달되는 것을 보여준다. 좌측 Mono address [00:14](k)와 우측 Color address [03:17](l)을 비교하면 둘 다 15개의 라인으로 구성되어

있으며, 번지만 다를 뿐이다. 컬러화 하는데 있어 흑백 데이터가 한 번 입력될 때마다 8번의 변화과정을 거쳐야 하므로 Counter(h)를 통하여 3개의 어드레스인 A0, A1, A2가 순차적으로 증가가 되고 Barrel shifter(g)에서는 S0, S1, S2에 따라 Data capture(f)에 저장된 8개의 데이터 중에서 하나를 선택하여 Multiplex(c)에 전달을 한다. Multiplex(c)에서는 전경색(Foreground color)과 배경색(Background color)을 선택해서 Color decoder(d)로 색상정보를 전달한다. Color decoder(d)에서는 입력이 4개(16 Color) 출력이 16개(65535 Color)로 구성되어 있으므로 65535 Color 중에서 16 Color를 사용자 임의로 선택할 수 있도록 구성하였다. Address reduction ROM(a)에서는 Color palette ROM(b)의 용량을 1/16 로 감소하기 위한 것이다. 입력 어드레스는 15개인데 비해서 출력은 11개로 감소하였다. 이것은 Color palette ROM(b)에서 중복되는 색상 정보를 최적화하기 위하여 어드레스를 감소한 것이다. Color

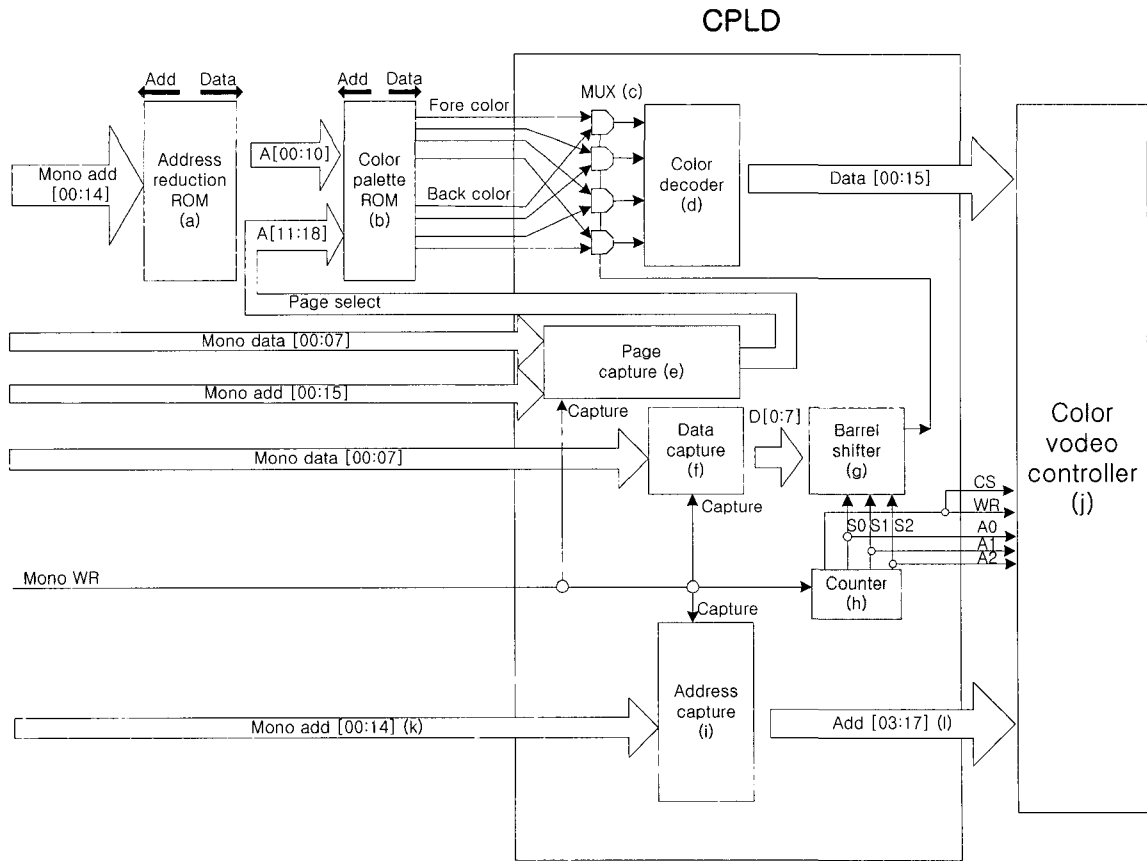


그림 12. 전체 구성도
Fig. 12. Block diagram of total system.

palette ROM(b)에서 입력 어드레스는 총 19개이며, 그 중에서 하위 11개와 상위 8개가 분리되어 사용된다. 하위 어드레스는 한 화면을 구성하는 절대 번지이며, 상위 8개의 어드레스는 256 페이지에 해당되는 상대 어드레스를 나타낸다.

흑백 데이터가 컬러 데이터로 변환되는 과정에 있어 화면을 나타내기 위해서는 가장 먼저 흑백 장치에서는 해당 페이지 정보를 보내야 한다. Page capture(e), Data capture(f), Address capture(i)에서는 화면처리에 필요한 정보를 Mono WR 제어에 따라서 임시로 저장한다. 실제 하드웨어 구성으로는 D-FF(D flip-flop)을 사용하고 있다. Counter(h)는 Mono WR 신호가 검출이 되면 동작하기 시작해서 S0, S1, S2 가 모두 카운트한 후에는 동작을 멈추고 다시 대기 상태로 된다.

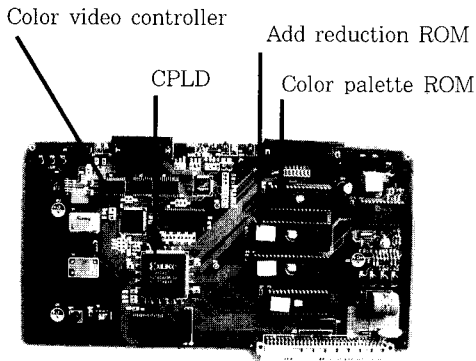


그림 13. Mono to Color 변환 보드.
Fig. 13. Monochrome to Color transformation board.

<그림 13>은 본 연구를 바탕으로 실제의 제품을 만든 것이며, 흑백으로부터 컬러로 실시간에 변환하는 보드를 보여준다. Address reduction ROM은 Data bus가 16개이며 Color palette ROM은 Data bus가 8개임을 시각적으로 알 수 있다. CPLD는 Xilinx 95288HQ208를 사용하였다.

V. 성능비교

<표 2>에서는 DSP(TMS320C32)를 이용한 소프트웨어 방식과 CPLD(95288HQ208)를 이용한 하드웨어 방식 두 가지를 해상도에 따라서 화면 Display 속도를 기록하였다. 결과적으로 CPLD 방식이 DSP 방식보다 대략 6배 정도의 빠른 속도로서 흑백데이터를 컬러데이

터로 변환시킬 수 있음을 알 수 있다. 또한 CPLD와 ROM의 성능이 곧 칼라변환의 시간요인으로 작용한다.

표 2. DSP와 CPLD 방식의 성능 비교
Table. 2. Performance comparison of DSP and CPLD method

구분	DSP(S/W 방식)	CPLD(H/D 방식)
320×240	0.55 Sec	0.09 Sec
640×400	1.82 Sec	0.31 Sec
800×600	3.42 Sec	0.58 Sec



그림 14. 시스템의 구성.
Fig. 14. Outlook of the designed system.

<그림 14>는 기존의 흑백 화면표시장치를 컬러 화면 표시장치로 교체한 제품의 동작 상태를 보여준다. 640×400의 해상도에서 한 화면 전체를 1회 디스플레이 하는 데 걸리는 시간은 대략 0.31초이며, 흑백 장치의 출력을 실시간에 컬러로 변환할 수 있었다. 소프트웨어 처리 방식인 DSP (Digital Signal Processor)를 사용하여 컬러 변환 실험 결과 한 화면 전체를 띄우는 시간이 대략 1.82초 정도의 시간이 소요되어 실시간 변환시스템으로 사용할 수 없음을 확인하였다. 결론적으로 기존의 흑백 표시장치를 컬러로 변환함에 있어서 실시간 성능을 확보하기 위해서는 CPLD를 사용하는 하드웨어 변환 방식이 유일한 해가 된다는 것을 알 수 있었다.

VI. 결 론

DSP(TMS320C32)를 사용하여 흑백 데이터를 컬러 데이터로 변환하는 장치를 설계 후, 640×400 해상도의 화면을 나타내기 위한 시간이 1.8초 정도로 비교적 길게 소요되어 실시간 처리가 불가능함을 인식하였다. 또한, 주변 회로가 복잡해짐에 따라 제품의 단가가 올라가는 것을 주지하게 되었다. 이러한 문제점을 해결하기 위하여 흑백 데이터를 실시간적으로 컬러 데이터로 변환하는 알고리즘을 CPLD를 사용하여 하나의 칩으로 제작해야 할 것으로 판단하고 이를 설계 및 제작하였다. 설계의 주안점은 중복된 색상을 축소하는 알고리즘을 정립하고, 결과 값을 ROM화 시켜 Color palette ROM의 용량을 1/16로 축소하는 것과 변환 알고리즘을 하드웨어적인 기법을 적용하여 실시간 처리하는 것이다.

현재 구현된 것은 640×400 해상도와 65535 컬러 중에서 16 컬러를 선택할 수 있고, 텍스트 모드로 사용하는 것으로 제한되어 있지만 응용 범위에 따라 더 넓은 해상도와 더 많은 컬러를 사용할 수 있다.

개발한 시스템을 산업현장에 널리 사용되고 있는 사출성형기에 적용한 결과 실시간 빠른 컬러 화면을 확인할 수 있었다.

차후에 진행할 연구는 모든 흑백 데이터를 텍스트 모드뿐만 아니라 그래픽 모드까지 실시간적으로 컬러로 변환하는 기법을 연구할 것이다.

참 고 문 헌

- [1] 심수석, "PWM 컬러 STN-LCD 제어기 설계 및 FPGA 구현", P.N.U., Feb., 1996
- [2] Fast EP-ROM 홈페이지, <http://www.st.com>
- [3] M. F. Tasi and H. C. Chen, "Design and Implementation of a CPLD-Based SVPWM ASIC for Variable-Speed Control of AC Motor Drives", IEEE. International Conference on, Power Electronics and Drive Systems, vol. 1, pp. 322-328, Jan., 2001.
- [4] C. Yu and H. Yu, "The design of general-purpose parallel interface based on CPLD", IEEE. 4th International Conference on, ASIC 2001, pp. 526-529, 2001.
- [5] David VAN Bout, "FPGA DESIGN 이론 및 실습", 홍릉과학출판사, 2000
- [6] 이준성, 서강수, "Xilinx Foundation을 이용한 디지털 시스템 설계", 복두출판사, 2001
- [7] W. Gunther, R. Drechsler, "Performance driven optimization for MUX based FPGAs", IEEE. 2001 Fourteenth International Conference on, VLSI Design, pp. 311-316, 2001.
- [8] Ting Wu, Chi-Ying Tsui, M. Hamdi, "A 2Gb/s 256*256 CMOS crossbar switch fabric core design using pipelined MUX", IEEE. International Symposium on, ISCAS 2002, Circuits and Systems, vol. 2, pp. 568-571, 2002.
- [9] Y. Fuji, J. O'Neill, P. Larsson, D. Inglis, J. Othmer, "A 1.5V 86mW/ch 8-channel 622-3125Mb/s/ch CMOS serdes macrocell with selectable MUX/DEMUX ratio", IEEE. 2002 International Solid-State Circuits Conference, Digest of Technical Papers, vol. 2, pp. 48-396, 2002.
- [10] R. Bakalash, Xu. Zhong, "A barrel shift microsystem for parallel processing", IEEE. Proceedings of the 23rd Annual Workshop and Symposium, Micro programming and Microarchitecture. Micro 23. Workshop on, pp. 223-229, 1990.

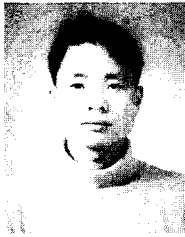
저 자 소 개



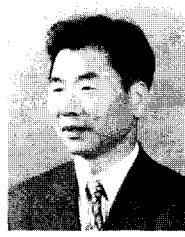
尹宰茂(正會員)
 2001년 2월 : 경성대학교 전기전자 공학과 졸업(공학사). 2001년 3월~현재 : 부산대학교 대학원 전자공학과 석박사통합과정 재학 중. <주관심분야 : CPLD 설계, 이동매니플레이터 등>



陳台錫(正會員)
 1998년 : 진주산업대학교 졸업(공학사). 2000년 : 부산대학교 대학원 졸업(공학석사). 2003년 8월 : 부산대학교 대학원 졸업(공학박사). <주관심분야 : 이동로봇 주행제어, 다중센서융합 등>



姜雄紀(正會員)
 2001년 2월 : 진주산업대학교 졸업(공학사). 2002년 3월~현재 : 부산대학교 대학원 전자공학과 석사과정 재학 중. <주관심분야 : DSP 프로세서 제어>



李章明(正會員)
 1992년~현재 : 부산대학교 전자공학과 교수. 2001년 5월~2003년 8월 : 컴퓨터 및 정보통신 연구소 소장