2003년도 대한전자공학회 하계종합학술대회 논문집 제26권 제1호

디스플레이 테스트를 위한 패턴 생성 회로 설계

조 경연
전자부품연구원 SoC 연구센터
전화 : 031-610-4054 / e-mail: choky@keti.re.kr

Design of Pattern Generation Circuit for Display Test

Kyoung-Youn Cho
SoC Research Center, Korea Electronics Technology Institute

Abstract

Now a days, many different kinds of display technologies such as Liquid Crystal Display (LCD), Organic Light Emitting Diode (OLED), and Liquid Crystal On Silicon (LCOS) are designed. And these display technologies will be used in many application products like High Definition Televisions (HDTVs) or mobile devices. In this paper, pattern generation circuit for display test is proposed. The proposed circuit will be embedded in the control circuit of display chip. Two differenct kinds of patterns is generated by the circuit. One is block pattern for color test, and the other is line pattern for pixel test. The shape of test pattern is determined by the values of registers in pattern generation circuit. The circuit is designed using Verilog HDL RTL code.

Ⅰ. 서론

현재 Liquid Crystal Display (LCD), Organic Light Emitting Diode (OLED), Liquid Crystal On Silicon (LCOS) 등 다양한 디스플레이 기술이 개발되고 있으며, 이러한 디스플레이 기술들은 향후 High Definition Television (HDTV) 및 모바일 디바이스를 비롯한 다양한 응용 제품 시장에 널리 활용될 것으로 예상된다. 이러한 디스플레이 판넬 제작 후에는 디스플레이 판넬의 컬러 및 동작을 검증하기 위한 패턴을 생성하는 회로가 컨트롤러 회로에 내장되어 판넬 테스트를 용이하게 하는 것이 필요하다[1]. 제안하는 회로는 디스플레이의 컨트롤 회로에 내장되어 외부의 입력 없이 테스트 동작 모드에서 자동으로 다양한 컬러의 블록 패턴 및 라인 패턴을 생성할 수 있다. 패턴 생성회로가 생성하는 블록 패턴 및 라인 패턴은 레지스터 값에 의하여 결정된다. 제안하는 회로는 Verilog HDL을 이용하여 Register Transfer Level (RTL)로 설계되었으며, ALTERA Flex 10K 라이브러리를 이용하여 합성하여 검증하였다. Ⅱ장에서는 패턴 생성 알고리즘을 제시하고, Ⅲ장에서 회로 설계에 대한 내용을 기술한다. 그리고, Ⅳ장에서 본 논문의 결론을 맺는다.

Ⅱ. 패턴 생성 알고리즘

제안하는 알고리즘이 생성하는 패턴은 블록 테스트 패턴과 라인 테스트 패턴으로 분류된다.

2.1 블록 테스트 패턴 생성 알고리즘

본 논문에서는 레지스터의 값만 설정함으로써 디스플레이에 원하는 모양의 테스트 패턴을 출력하게 하는 회로를 제안한다. 할당하는 레지스터는 2가지로 분류되는데 컬러 값을 결정하는 C 레지스터와 블록 패턴을 결정하는 레지스터이다. 블록 패턴을 결정하는 레지스터는 다시 수평 블록의 폭을 결정하는 H 레지스터와 수직 블록의 폭을 결정하는 V 레지스터로 분류된다. 한 프레임의 컬러 및 블록의 폭의 수는 레지스터의 수에 따라 결정되며, 주기적으로 반복된다. 예를 들어 4개의 C 레지스터와 2개의 H 레지스터, 그리고, 2개의

V 레지스터를 사용할 경우 그림 1과 같은 테스트 패턴을 생성할 수 있다. 그림 1에서 H1과 H2는 H 레지스터이고, V1과 V2는 V 레지스터이다. 그리고 숫자 1, 2, 3, 4 는 C 레지스터에 저장된 컬러를 나타낸다. 즉, 테스트 패턴의 블록 모양은 H 레지스터와 V 레지스터에 의해서 결정되고, 각 블록의 컬러는 C 레지스터의 값을 주기적으로 반복함으로 결정된다. 블록 모양의 폭의 수를 N개로 할 경우에는 H 또는 V 레지스터의 숫자를 N으로 증가시킨다. 그림 2에 본 논문에서 블록 모양의 테스트 패턴을 생성하기 위하여 제안하는 방법의 동작 흐름도를 제시하였다. CSel, HSel, VSel은 레지스터를 지시하는 인덱스 값으로, CSel은 C 레지스터를 지시하고, HSel은 H 레지스터를 지시한다. 그리고, VSel은 V 레지스터를 지시한다. 초기값은 모두 각 레지스터의 첫 번째 레지스터를 지시하도록 한다. 회로가 테스트 모드로 동작하지 않을 경우, 모든 인덱스 레지스터의 값을 초기값으로 설정한다. 회로가 테스트 모드로 동작을 하면, 다음의 절차에 따라서 동작을 한다. 수평 주사가 끝나지 않았을 경우 현재의 수평 블록에 해당하는 픽셀을 모두 디스플레이 하였나를 조사한다. 이것은 현재의 HSel 인덱스가 지시하는 H 레지스터에 저장되어 있는 값만큼 픽셀을 디스플레이 하였는지를 조사함으로써 판단 가능하다. 모두 디스플레이 하였을 경우 CSel과 HSel 이 다음 레지스터를 지시하도록 하고 새로운 컬러 값을 디스플레이하고, 그렇지 않은 경우 현재의 컬러 값을 디스플레이 한다. 수평 주사가 끝났을 경우 수직 주사가 끝났는지를 조사하고, 수직 주사가 끝나지 않았을 경우 현재의 수직 블록에 해당하는 수평 주사선을 모두 디스플레이 하였나를 조사한다. 이것은 현재의 VSel 인덱스가 지시하는 V 레지스터에 저장되어 있는 값만큼 수직 라인을 디스플레이 하였는지를 조사함으로써 판단 가능하다. 만일 현재의 블록에 해당하는 수평 주사선을 모두 디스플레이 하였다면, VSel, CSel 인덱스를 다음 값으로 할당하고, HSel 은 초기값으로 할당하며, CselSave 에 다음 CSel 값을 할당한다. 현재의 수직 블록에 해당하는 수평 주사선 디스플레이가 모두 끝나지 않았을 경우 CSel에 CselSave 값을 할당하고 HSel에 초기값을 할당한다. 수직 주사가 끝났을 경우 모든 인덱스 값을 초기값으로 할당한다. 그리고, 다시 이 과정을 처음부터 반복하게 된다.

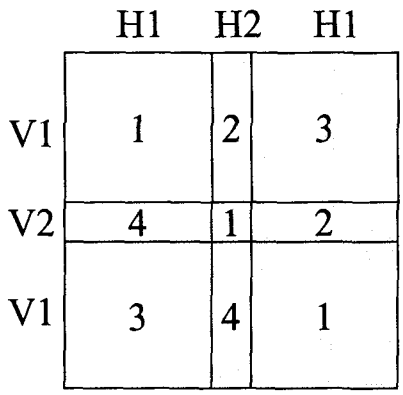

그림 1. 테스트 패턴 생성 예

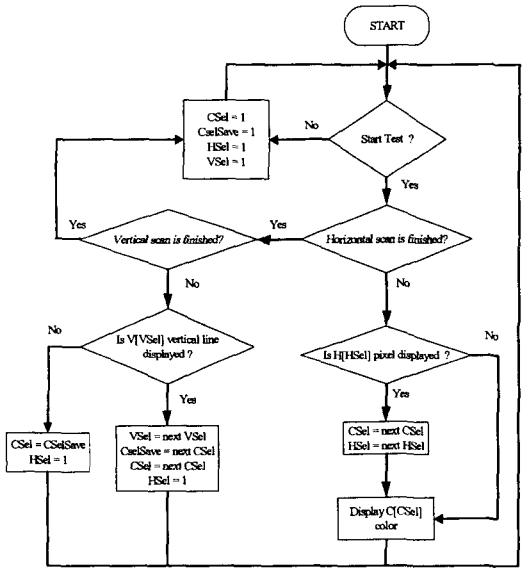

그림 2. 블록 테스트 패턴 생성 동작 흐름도

2.2 라인 테스트 패턴 생성 알고리즘

그림 3에 본 논문에서 제안하는 방법으로 생성할 수 있는 라인 테스트 패턴의 종류를 제시하였다. 그림 3과 같이 본 논문에서 제안하는 방법으로 생성 가능한 라인 테스트 패턴은 4 종류인데, Vertical 라인 패턴, Horizontal 라인 패턴, Left Bias 라인 패턴, Right Bias 라인 패턴으로 구성된다. 그림 3에서 Start point로부터 시작하여 n번째 픽셀의 수평 좌표 값을 $H(n)$이라 하고 수직 좌표 값을 $V(n)$이라고 하면 각각의 라인 테스트 패턴에 대하여 $(n+1)$번째 라인 테스트 패턴의

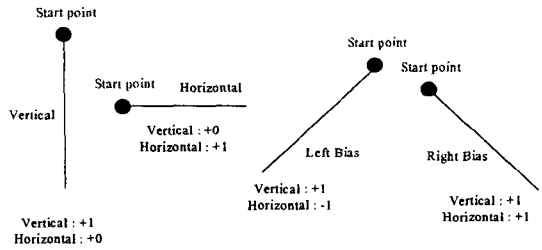

그림 3. 본 논문에서 제안하는 방법으로 생성 가능한 라인 테스트 패턴

좌표값은 식(1)-식(4)를 이용하여 구할 수 있다.

Vertical 라인 패턴 :

$$V(n+1) = V(n) + 1,\ H(n+1) = H(n) \quad (1)$$

Horizontal 라인 패턴 :

$$V(n+1) = V(n),\ H(n+1) = H(n) + 1 \quad (2)$$

Left Bias 라인 패턴 :

$$V(n+1) = V(n) + 1,\ H(n+1) = H(n) - 1 \quad (3)$$

Right Bias 라인 패턴 :

$$V(n+1) = V(n) + 1,\ H(n+1) = H(n) + 1 \quad (4)$$

본 논문에서는 식(1)-식(4)의 특성을 이용하여 라인 테스트 패턴의 Start point 값과 라인의 길이 값만을 레지스터에 할당함으로써 라인 테스트 패턴을 생성하는 구조를 제안한다. 그림 4에 본 논문에서 라인 테스트 패턴을 생성하기 위하여 제안하는 방법의 흐름도를 제시하였다. 그림 4에서 Hcnt와 Vcnt는 현재 디스플레이하는 픽셀의 수평 좌표와 수직 좌표이다. Vertical은 라인의 수직 좌표를 표시하는 변수이고, Horizontal은 라인의 수평 좌표를 표시하는 변수이다. 또한 Length는 라인의 길이를 표시하는 변수이다. 회로가 테스트 모드로 동작하기 전에는 초기값으로 Start point의 수평 좌표와 수직 좌표를 레지스터로부터 Horizontal, Vertical 변수에 저장한다. 만일 현재 디스플레이하고 있는 픽셀의 좌표 (Vcnt, Hcnt)가 (Vertical, Horizontall)과 일치하면, 현재의 디스플레이 위치(Vcnt, Hcnt)에 라인 테스트 패턴의 컬러를 출력하고, 식(1)-식(4)에 따라 Vertical 변수 값과 Horizontal 변수 값을 Update한다. 그리고, Length 변수 값을 1 감소한다. 이때 Length가 0

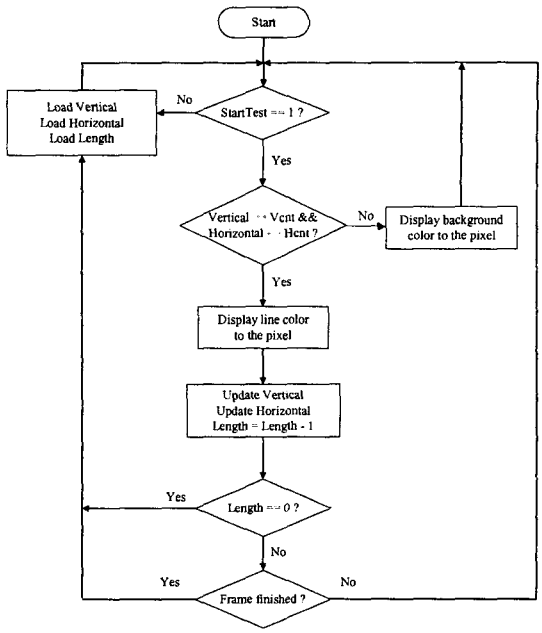

그림 7. 라인 테스트 패턴 생성 동작 흐름도

이거나, 한 프레임 디스플레이가 끝났으면 Vertical, Horizontal, Length 변수 값을 초기값으로 할당한다. 만일 (Vcnt, Hcnt)와 (Vertical, Horizontal)이 일치하지 않는 다면 현재의 디스플레이 위치(Vcnt, Hcnt)에 배경 컬러를 디스플레이 한다. 만일 (Vertical, Horizontal) 좌표가 디스플레이 영역을 벗어나게 되면, 이 좌표는 (Vcnt, Hcnt)와 일치할 수 없기 때문에 자동적으로 배경 컬러를 디스플레이 하게된다.

Ⅲ. 회로 설계

본 논문에서는 Verilog HDL를 이용하여 패턴 생성 회로를 설계하였다. 설계된 회로는 ModelSim[2]을 이용하여 functional 시뮬레이션을 수행하여 function을 검증하였다. Function 검증은 Verilog testbench에서 출력 RGB 값을 파일로 출력하여, 예상한 출력 패턴과 비교를 하여 검증하였다. Function 검증 후 Synplify Pro 툴[3]을 이용하여 ALTERA EP20K100EQC240-3 디바이스[4]를 타켓으로 하여 합성하였고, Quartus 툴을 이용하여 P&R 후 standard delay file (sdf)을 추출하고, ModelSim을 이용하여 타이밍 시뮬레이션을 수행하여 검증하였다. 설계된 회로는 디바이스의 약 4%의 자원을 사용하였고, 45 MHz의 클럭 속도로 동작이 가능하였다. 그림 5에 본 논문에서 적용한 설계 플로우를 제시하였고, 그림 6에 합성한 회로의 블록 다이

어그램을 제시하였다. 사용한 레지스터는 총 54 bytes 로 각 레지스터의 기능은 표 1과 같다.

Ⅳ. 결론

본 논문에서는 외부의 입력 없이 디스플레이를 테스트하기 위한 다양한 테스트 패턴을 생성하는 내장 회로를 제안하였다. 외부의 입력이 필요 없기 때문에 손쉽게 디스플레이를 테스트 할 수 있으며, 레지스터 값만 변경함으로써 다양한 블록 테스트 패턴을 생성할 수 있고, 각 블록 패턴에 다양한 컬러를 할당하여 테스트를 할 수 있다. 또한 디스플레이의 픽셀을 테스트하기 위한 라인 테스트 패턴을 임의의 위치에 출력할 수 있기 때문에 디스플레이 판넬 제조 후 손쉽게 테스트를 할 수 있다. 본 논문에서 제안하는 방법은 디스플레이의 제어 회로에 내장되어 동작할 수 있으며, 서로 다른 특성을 가지는 디스플레이를 효율적으로 테스트하기 위한 다양한 테스트 패턴을 레지스터 값만을 할당함으로써 쉽게 생성할 수 있다는 장점이 있다.

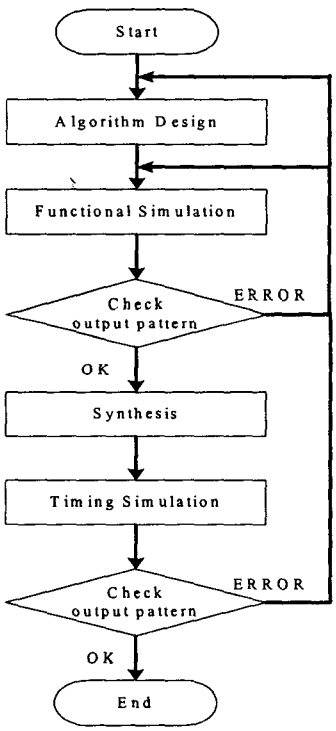

그림 5 설계 플로우

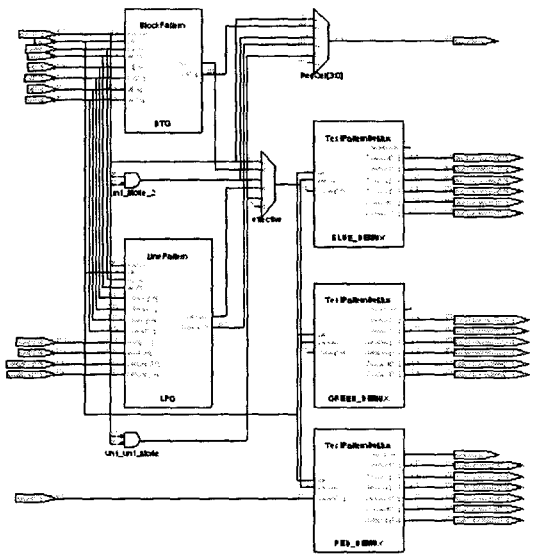

그림 6. 패턴 생성회로의 블록다이어 그램

표 2. 레지스터 할당

Register	Block mode	Line mode
Hf, Hs (8 bits)	Horizontal width	Hf[7:4] : Mode {Hf[2:0], Hs[7:0]} : Horizontal start point
Vf, Vs (8 bits)	Vertical width	{Vf[2:0], Vs[7:0]} : Vertical start point
RegREG (16x8bits)	Values for Red color	RedREG[0] : Value for background color RedREG[1] : Value for line color
GreenREG (16x8bits)	Values for Green color	GreenREG[0] : Value for background color GreenREG[1] : Value for line color
BlueREG (16x8bits)	Values for Blue color	BlueREG[0] : Value for background color BlueREG[1] : Value for line color
Length_H, Length_L (8bits)	-	{Length_H[7:0], Length_L[7:0]} : Lenght of line

참고문헌

[1] e-MDT(http://www.e-MDT.com). *LCOS Display Chip Set (MD1920) Technical report*, Nov. 2002

[2] http://www.model.com

[3] http://www.synplicity.com

[4] http://www.altera.com