

LCD 신호 전송을 위한 엔코딩 알고리즘

박병기, 최철호, 박진성, 최명렬
한양대학교 제어계측공학과

Encoding Algorithm for LCD Interconnection Line

Byung Ki Park, Chul Ho Choi, Jin Sung Park, Myung Ryul Choi
Dept. of Control & Instrumentation Engineering, Hanyang University

요약

기존의 평판 디스플레이 비디오 신호 전송 방식은 그래픽 어댑터에 저장된 비디오 데이터를 DA 컨버터를 통해 아날로그 신호를 전송한 다음 평판 디스플레이 패널에서 다시 AD 컨버터를 이용해 디지털 신호로 복원한 다음 평판 디스플레이용 컨트롤러에 입력되었다. 아날로그 신호를 보내는 것은 AD/DA 컨버터를 이점으로 사용할 뿐 아니라, 연결선을 길게 하는 것이 어렵고 노이즈 인해 데이터의 수치가 변할 우려가 있다. 패널의 고면적화와 고해상도의 기술 발달로 SXGA급 및 UXGA급 신호에서는 높은 클럭 주파수와 긴 연결선에서도 안정된 신호 전송을 할 수 있는 VESA FPD1 규격에 맞는 인터페이스는 노이즈에 강하고 전자파 발생 억제 및 저전력으로 동작을 한다. 본 논문에서는 인터페이스 신호 전송 방법을 연구하여 신호의 변화를 줄여서 전자파 발생 억제 및 저전력에 도움을 주는 비디오 신호 엔코딩 알고리즘을 제안하였다.

1. 서론

정보 산업의 발달과 과학 기술의 발달로 컴퓨터 관련 산업이 급격히 발달하고 있다. 과학 기술이 발달함에 따라 종래의 기능 위주의 개발에서 발전되어 인간 중심의 개발에 더욱 차이를 하고 있다. 컴퓨터 영상 표시 장치 분야에서도 CRT(Cathode Ray Tube) 모니터의 단점을 보완하고 인간에게 저해 요소가 적은 LCD(Liquid Crystal Display)의 개발이 활발히 이루어지고 있다. CRT와 LCD의 차이점을 간단히 살펴보면 CRT는 화면 크기에 비해 부피가 크고 전력 소모가 크며 아날로그 방식으로 되어 있는 반면, 우수한 해상도와 색상 표시 능력을 가지고 있다. LCD는 CRT와는 다르게 화면 크기에 비해 부피가 작고 전력 소모가 적으며 디지털 방식으로 되어 있고 현재까지는 해상도와 색상 표시 능력은 CRT에 비해 뒤떨어지지만, LCD는 CRT보다 전자파 발생이 적으며 디지털 신호를 이용하기 때문에 CRT에서처럼 그래픽 어댑터에서 디지털 신호를 아날로그 신호로 변환시켜주는 DA 컨버터가 필요 없게 된다.

24비트의 트루컬러의 이미지 데이터를 전송할 경우 24개의 데이터 선과 주파수 동기화를 위한 신호선 및 클럭을 포함해서 27개의 버스선이 필요하게 된다. 버스선 수의 증가로 인해 그

래픽 어댑터와 LCD 패널간의 연결 라인(interconnection Line)에서 전자파 발생 및 과도한 전선의 소비가 문제 시 되었는데 이에 대한 해결책으로 테이터를 직렬화(serialize) 시켜서 전송시키는 방안이 제안되었고, VESA FPD1-2(Flat Panel Display Interface)로 규정되었다. FPD1-2에 따르면 각 신호는 LVDS(Low Voltage Differential Swing) 신호로 전달되며, RGB(Red Green Blue)와 클럭의 4개의 신호가 각 신호마다 쌍(pair)을 이루어서 전송되도록 되어있다. 이런 방법으로 인해 버스선의 수는 8선으로 줄었으며 전자파 발생을 줄이는 데에도 크게 기여하였다[3].

LCD에서 사용하는 화상 데이터 신호의 특성은 RGB 신호를 사용하여 각각의 컬러가 한 픽셀 당 8비트의 정보를 가지고 있으며 컴퓨터 모니터용으로 주로 사용하는 이미지의 경우에는 같은 색이 연속적으로 반복되는 빈도가 매우 높다. 비록 시진과 같은 천연색의 이미지라 할 지라도 RGB 각각으로 살펴보면 같은 색의 연속이 빈번할 뿐 아니라 다를 지라도 다소 작은 차의 변화를 가진 데이터가 많다.

본 논문에서는 LVDS 신호를 이용해서 직렬로 전달되는 연결 라인에서 전송되는 비디오 신호의 특성을 분석하고 이를 이용하여 데이터 전송 시 신호의 변화(transition)를 줄임으로써 저주파의 데이터로 만들어 신뢰성 있는 비디오 신호를 위한 엔코딩 알고리즘을 제안하였다.

2. 기존의 알고리즘

현재 사용되고 있는 알고리즘은 DC-balanced code[1]를 이용하여 변화를 줄이는 방법으로 8비트의 정보에 2비트의 제어 비트를 추가하여 데이터 변화를 줄이는 대신에 데이터가 10비트로 늘어나는 점이 있다

그림 1과 같이 '1'의 수가 4보다 작을 경우에는 XOR 연산 루틴의 결과를 사용하면서 추가로 '1'을 붙이고, '0'의 수가

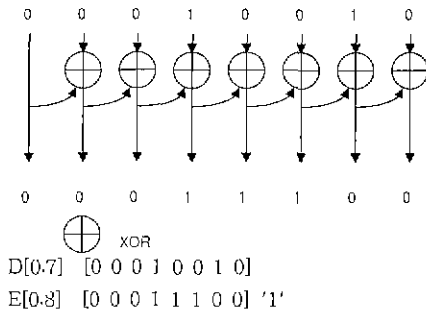


그림 1 Chained XOR

4보다 작을 경우는 그림 2과 같이 XNOR 연산 루틴을 통해 계산된 결과에 '0'을 추가해서 다음 블록으로 넘어간다

그림 3의 DC Balanced Block에서는 바로 전의 데이터와의 불균형을 비교하여 9비트의 데이터를 반전할 것인지 하지

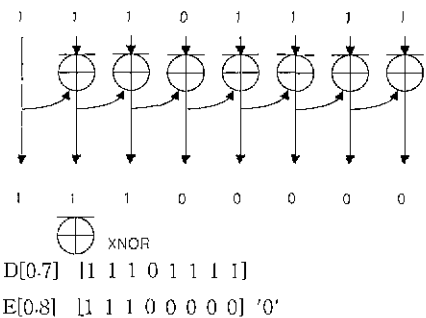


그림 2 Chained XNOR

않을 것인지 판단하여 반전할 경우는 '1'을 추가하고, 하지 않을 경우는 '0'을 추가 시켜서 최종 엔코딩한 값을 얻게 된다
이 알고리즘은 직렬의 데이터를 전송할 때 최대 22%까지 변화를 줄일 수 있다[2].

3. 제안한 알고리즘

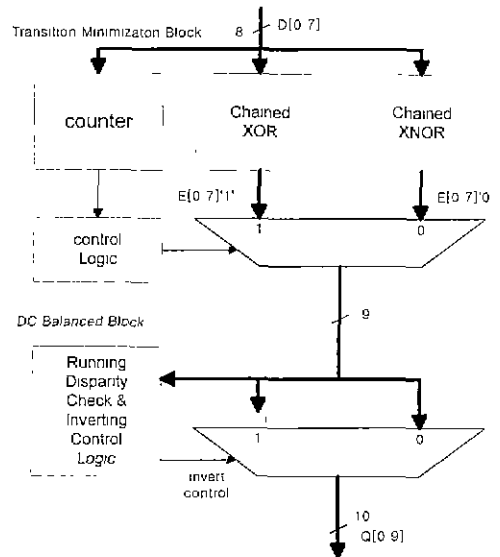


그림 3 DC balanced coder 블록도

본 논문에서 제안한 알고리즘은 이미지 데이터의 특성이 같은 색의 반복적인 요인을 고려하여 현재 화소 데이터와 이전 화소 데이터의 차를 이용하였다.

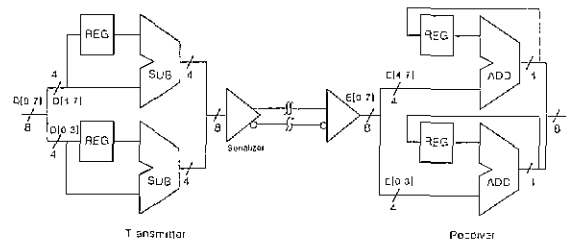


그림 4 제안한 엔코딩 알고리즘 블록도

그림 4에서와 같이 입력된 8비트의 데이터를 상위 4비트, 하위 4비트로 분리하여 각각을 현재 값에서 이전 값이 저장된 REG(레지스터)의 값으로 뺀다. 두개의 차를 다시 8비트의 데이터로 상위, 하위를 합침 다음 직렬화 드라이버(Serialize Driver)를 통해 전달한다 수신단에서 받은 데이터는 이전 값을 저장하고 있는 REG 값과 더하여 현재의 값을 복원에 낸다. 송신단에 SUB(감산기)가 있기 때문에 그 결과가 음수일 수 있기 때문에 부호를 나타내는 비트가 필요한 문제가 있으나 실제로는 그래픽 데이터는 0~255 사이의 양의 정수이고 상위 4비트로 나누어도 0~15까지의 양의 정수이므로 표 1에서와 같이 차가 음수일 때는 2의 보수를 취하는 SUB를 사용하면 추가하는 비트 없이 정보를 잃지 않고 전달 할 수 있다 표 1에서 차에 대한 항목은 10진수이며, 현재 값-이전 값의 항목은 16진수, 전달값은 2진수로 나타내었다. 이전 값은 REG(레지스터)에 저장되어 있는 값이고, 현재 값은 현재 클

4	현재 값(n)-이전 값(n-1)																신뢰도
0	0-0	1-1	2-2	3-3	4-1	5-5	6-6	7-7	8-8	9-9	a-a	b-b	c-c	d-d	e-e	f-f	0000
1	1-0	2-1	3-2	4-3	5-1	6-5	7-6	8-7	9-8	a-9	b-a	c-b	d-c	e-d	f-e		0001
2	2-0	3-1	4-2	5-3	6-1	7-5	8-6	9-7	a-8	b-9	c-a	d-b	e-c	f-d			0010
3	3-0	4-1	5-2	6-3	7-1	8-5	9-6	a-7	b-8	c-9	d-a	e-b	f-c				0011
4	4-0	5-1	6-2	7-3	8-1	9-5	a-6	b-7	c-8	d-9	e-a	f-b					0100
5	5-0	6-1	7-2	8-3	9-1	a-5	b-6	c-7	d-8	e-9	f-a						0101
6	6-0	7-1	8-2	9-3	a-1	b-5	c-6	d-7	e-8	f-9							0110
7	7-0	8-1	9-2	a-3	b-1	c-5	d-6	e-7	f-8								0111
8	8-0	9-1	a-2	b-3	c-1	d-5	e-6	f-7									1000
9	9-0	a-1	b-2	c-3	d-1	e-5	f-6										1001
10	a-0	b-1	c-2	d-3	e-1	f-5											1010
11	b-0	c-1	d-2	e-3	f-1												1011
12	c-0	d-1	e-2	f-3													1100
13	d-0	e-1	f-2														1101
14	e-0	f-1															1110
15	f-0																1111
-1	0-1	1-2	2-3	3-1	4-5	5-6	6-7	7-8	8-9	9-a	a-b	b-c	c-d	d-e	e-f		1111
-2	0-2	1-3	2-4	3-5	4-6	5-7	6-8	7-9	8-a	9-b	a-c	b-d	c-e	d-f			1110
-3	0-3	1-4	2-5	3-6	4-7	5-8	6-9	7-a	8-b	9-c	a-d	b-e	c-f				1101
-4	0-4	1-5	2-6	3-7	4-8	5-9	6-a	7-b	8-c	9-d	a-e	b-f					1100
-5	0-a	1-b	2-c	3-d	4-b	5-a	6-b	7-c	8-d	9-e	a-f						1011
-6	0-b	1-c	2-d	3-e	4-c	5-b	6-c	7-d	8-e	9-f							1010
-7	0-1	1-2	2-3	3-4	4-b	5-c	6-d	7-c	8-f								1001
-8	0-8	1-9	2-a	3-b	4-c	5-d	6-e	7-f									1000
-9	0-9	1-a	2-b	3-c	4-d	5-e	6-f										0111
-10	0-a	1-b	2-c	3-d	4-e	5-f											0110
-11	0-b	1-c	2-d	3-e	4-f												0101
-12	0-c	1-d	2-e	3-f													0100
-13	0-d	1-e	2-f														0011
-14	0-e	1-f															0010
-15	0-f																0001

표 1 엔코딩 방법에서 모든 경우의 수

력에 입력되는 값을 의미한다. 동적하는 방법을 예를 들어 설명하면 다음과 같다 REG(레지스터)에 저장되어 있는 값이 3이고 현재 들어온 값이 8라면 현재 값(n)-이전 값(n-1)은 5이므로 전달되는 값은 '0101'이다. 수신측 REG(레지스터)에는 이전 데이터인 3이 저장되어 있으므로 이 값에 '0101'를 더하면 8이 되어서 데이터가 복원되었음을 알 수 있다 만약 레지스터에 있는 이전 데이터가 8이고 현재 들어온 데이터가 3이면 현재 값(n)-이전 값(n-1)은 -5가되어 전달값은 '1011'이 된다. 수신측의 레지스터에 저장되어 있는 8('1000')에 '1011'을 더하면 '10011'이 되는데 4비트 덧셈이기 때문에 '0011'만 이해가 되고 최상위 비트 '1'은 캐리(carry)가 되어 버려지게 된다 즉 '0011'은 16진수 3에 해당하므로 차가 음수인 경우에도 올바르게 전달되었음을 알 수 있다

두 번째 알고리즘은 상하위 4비트씩 나누지 않고 8비트 전체를 8비트 SUB를 통하여 인산하는 방식이다. 첫 번째 알고리즘은 4비트 연산만 하면 되기 때문에 8비트 연산보다 빠르게 계산할 수 있다

4. 시뮬레이션

PC를 이용하여 C 로써 실제 VGA급 24비트 컬러의 이미지를 사용하여 변화를 측정하였으며, Synopsys VHDL 틀을 이용하여 회로를 합성하였다.

		Bypass	AL 1	감소 %	AL2	감소 %
예1	R	1165649	702144	39.8	602220	48.3
	G	1125738	710057	37	610183	45.8
	B	1057703	670246	36.6	574356	45.7
예2	R	429618	140212	67.3	123614	71.2
	G	439898	145962	66.8	123614	71.9
	B	459167	149279	67.5	133411	70.9

표 2 시뮬레이션 결과



그림 5 예제 이미지 1

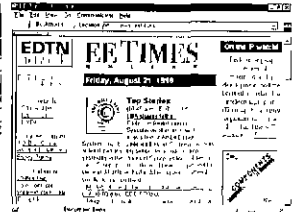


그림 6 예제 이미지 2

시뮬레이션에서 사진과 같은 이미지에서는 알고리즘 1은 37.8%, 알고리즘 2는 46.6%의 변화의 감소가 있었으며, 그림 6과 같은 인터넷 문서에 대해서는 알고리즘 1은 67.2%, 알고리즘 2는 71.3%의 변화를 감소시켰다

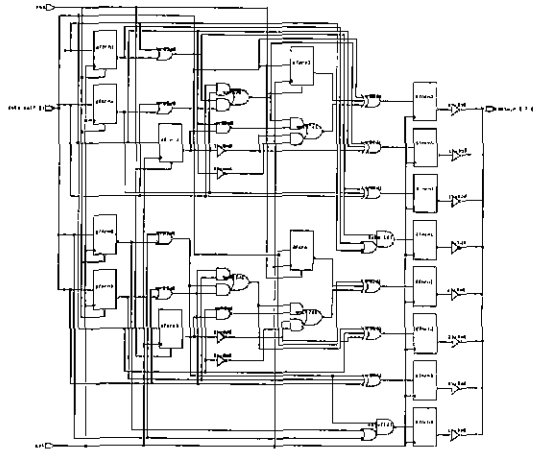


그림 7 VHDL을 이용한 회로 합성도

5. 결론

DC-Balanced code는 일반적인 신호에 대해서 변화를 줄여 주지만 비디오 신호에서는 신호의 특성상 다른 방법이 있는 것을 알 수 있었다 제안한 알고리즘 2가 알고리즘 1보다 성능이 우수하게 나왔지만 알고리즘 1은 4비트의 SUB와 ADD를 사용하기 때문에 빠르고 간단한 하드웨어로 구성할 수 있다.

참고문헌

[1] Kyeongho Lee, et al, High Speed Low EMI Digital Video Interface with Cable Deskewing and Transition minimizing Coding, Symposium on VLSI Circuit, 1997
 [2] Peter A Franasz, et al, Byte Oriented DC Balanced 8E/10B Partitioned Block Transmission Code, United States Patent 4486739, 1984
 [3] K. Lee, et. al, A Jitter-Tolerant 4.5Gb/s CMOS Interconnect for Digital Display, IEEE ISSCC98, pp 310-311, 1998