

LCD 구동 시스템에서 전력 소비 및 전자기 장애를 줄이기 위한 데이터 코딩 방법

(Data Coding Scheme to Reduce Power Consumption and EMI in LCD Driving Systems)

최 철 호 [†] 최 명 렬 ^{**}
(Chul-Ho Choi) (Myung-Ryul Choi)

요 약 본 논문에서는 LCD(Liquid Crystal Display) 컨트롤러와 LCD 구동 칩 사이의 신호 전송에서 전력 소비 및 전자기 장애(EMI)를 줄이기 위한 데이터 코딩 방법을 제안하였다. 제안한 방식은 컴퓨터의 일반적인 텍스트 화면에 대한 데이터 천이(transition)를 감소시키는 방식으로, 하드웨어 구조가 간단하여 실시간 처리가 요구되는 LCD 구동 시스템에 적용하기 용이하다.

제안한 방식의 성능 분석을 위해 컴퓨터를 이용한 시뮬레이션을 수행하였으며 기존의 코딩 방식과의 성능 비교 및 평가를 하였다. 시뮬레이션을 통해 제안한 방식이 기존 방식에 비하여 텍스트 화면과 그림 화면에 대해서 switching activity를 현저히 줄일 수 있음을 확인하였다.

Abstract We propose a data coding scheme for reducing power consumption and EMI in transmitting a sequence of data from LCD controller to LCD driver. The proposed coding scheme makes use of reducing data transitions in general text image of PC. It can be implemented with a little hardware and applied to the real-time applications of LCD driving system.

We have executed computer simulations of the proposed coding scheme and compared the results of the proposed scheme with those produced by the existing coding schemes. The proposed coding scheme, compared to the existing ones, reduces the switching activity significantly in both of text and picture images.

1. 서 론

정보화 산업의 발달은 오늘날 멀티미디어 시스템의 급속한 보급을 가져왔으며 인간과 기계 장치의 가교적인 역할을 담당하는 디스플레이 시스템의 중요성은 날이 증대되고 있는 실정이다. 최근 기존의 부피가 크고, 많은 전력을 소모하는 CRT(Cathode Ray Tube)를 대체할 수 있는 장치로서 FPD(Flat Panel Display)가 각광을 받고 있다. FPD는 CRT에 비하여 저전력, 경량, 평면성 등의 장점을 가지고 전세계 시장을 넓혀가고 있다. 1998년 140억 달러의 규모에서 2004년에는 260억

달러의 규모에 이를 전망이다. FPD에서 가장 큰 부분을 차지하는 컴퓨터 부분은 2004년에 LCD가 FPD 시장의 84%를 지배할 것으로 예상되며, 그 중 TFT LCD가 전체 LCD 시장에서 74% 이상을 점유할 것으로 전망되고 있다[1].

전자 기술의 발달로 인하여 디지털 시스템은 수백 MHz대의 처리 속도로 동작이 가능해졌지만, 여기서 야기되는 전력 소모와 EMI(Electromagnetic Interference)가 새로운 문제로 대두되고 있다. LCD 구동 시스템은 기존 CRT의 아날로그 방식이 아닌 디지털 방식으로 구동되고 있기 때문에 고해상도 및 대면적의 LCD는 처리할 데이터 양이 방대하다[2]. 즉, 24비트의 트루 컬러의 화면 데이터를 전송할 경우 기본적으로 24개의 데이터 선과 주파수 동기기를 위한 신호선 및 제어 신호를 포함해서 30여개의 버스선이 필요하게 된다. 하지만 LCD 패널 크기가 커지고 표현 가능한 해상도가 증가했

[†] 비 회 원 한양대학교 전자전기제어계측공학과
hbw@asic.hanyang.ac.kr

^{**} 종신회원 : 한양대학교 전자컴퓨터공학부 교수
choimy@asic.hanyang.ac.kr

논문접수 : 1998년 9월 20일

심사완료 : 2000년 10월 26일

지만, 그에 따른 LCD 구동 칩의 속도는 80MHz 대를 그치고 있는 실정이다. 따라서 SXGA(1280×1024)급 이상 해상도를 가진 LCD를 구동시키기 위해서는 적어도 100MHz의 처리 능력을 가진 구동 칩이 필요하다. 현재 많이 쓰이는 해결책으로 구동 칩의 데이터 버스를 두 배로 하여 구동시키는 방식을 채택하고 있다 (그림 1).

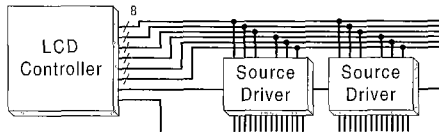


그림 1 LCD Controller와 구동 칩과의 인터페이스

본 논문에서는 LCD 구동 시스템에서 야기되는 소비 전력과 EMI를 줄이기 위한 방법으로 컴퓨터의 일반적인 텍스트 화면에 대한 데이터의 천이 특성을 고려하여 LCD 컨트롤러와 LCD 구동 칩 사이의 신호 전송에서 switching activity를 줄일 수 있는 병렬 signaling 방법을 제안한다. 2장에서는 디지털 시스템의 전력에 대해서 설명하고, 3장에서는 기존의 signaling 방법에 대하여 설명한다. 4장에서는 제안한 signaling 방법의 원리와 구성에 대해서 설명하고, 성능 평가를 위해 기존 방식과 제안한 방식을 다양한 실험 데이터를 가지고 성능 평가를 하며, 또한 제안한 방식의 특징과 응용분야에 대하여 논한다.

2. 디지털 시스템의 전력

TTL(Transistor-Transistor Logic), ECL(Emitter-Coupled Logic) 또는 GaAs회로에 비하여 CMOS (Complementary Metal Oxide Semiconductor) 회로의 소비 전력은 월등히 적기 때문에 디지털 회로에 널리 사용되고 있다. CMOS 회로에서 전력, 즉 단위 시간당 소비되는 에너지의 양은 정적인(static) 전력 소비와 동적인(dynamic) 전력 소비로 나눌 수 있다. 여기서 정적인(static) 전력 소비는 $nW(10^{-9}W)$ 급이므로 무시할 수 있다. 동적인(dynamic) 전력 소비는 식 (1)로 표현된다. 여기서 C_{loadi} 는 노드 i 의 부하 capacitance이고, V_{dd} 는 공급 전압을, f 는 동작 주파수를, P_{ti} 는 노드 i 의 activity factor를 나타낸다.

$$P = C_{loadi} \cdot V_{dd}^2 \cdot f \cdot P_{ti} \quad (1)$$

따라서, 저전력 회로를 구현하기 위해서는 C_{loadi} , V_{dd} , f 그리고 P_{ti} 를 최소화해야 한다. 여기서 V_{dd} 를 줄이면 전력 소비는 그의 제곱에 비례하여 줄기 때문에 가장

효과적인 접근 방법이다. 예를 들어, V_{dd} 를 5V에서 3V로 줄일 수 있다면 그에 따라 65%의 전력 소비를 감소시킬 수 있다. 하지만, MOSFET(MOS Field Effect Transistor) 지연 시간, t_d 는 반대로 증가하는 단점이 있다. 그리고 C_{loadi} 와 f 요소는 미리 결정되어 있는 경우가 일반적이며 이들 요소의 변경은 디지털 시스템에서 많은 비용을 초래한다. 반면 activity factor, P_{ti} 를 줄이는 방법은 저 전력 회로를 구현하는데 있어서 적은 하드웨어 비용으로 효과적인 결과를 얻을 수 있으며, 전하의 충전 및 방전에서 발생하는 EMI도 줄일 수 있는 방법이라고 할 수 있다[3].

또한, 디지털 시스템에서 소비되는 전력 중에서 적게는 10%, 많게는 80%가 I/O에서 소비되는 전력이다. 왜냐하면 I/O 핀의 capacitance 값이 칩 내부 capacitance 값보다 훨씬 크기 때문이다[4].

본 논문에서는 저전력 동작 및 EMI를 줄일 수 있는 방법으로 칩과 칩을 연결해주는 I/O 라인들의 switching activity를 줄일 수 있는 방식의 연구에 초점을 맞추었다.

3. 기존의 signaling 방법

Switching activity를 줄이기 위해서는 그림 2에 도시한 것처럼 엔코더, 디코더, 그리고 제어 라인과 같은 추가적인 하드웨어가 필요하다. 따라서 간단한 구조의 엔코더 및 디코더, 적은 수의 제어 라인으로 구현하는 것이 무엇보다 중요하다. 여기서 제어 라인 없이($m=n$) 원래의 I/O 핀의 수로 switching activity를 줄이기 위해서는 먼저 데이터들의 통계 분석이 선행되어야만 가능하다[3].

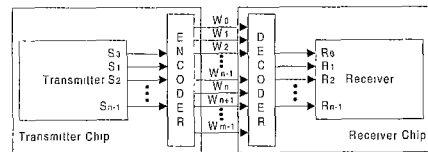


그림 2 디지털 시스템에서 switching activity를 줄이기 위한 추가적인 하드웨어

3.1 One-Hot 코딩

이 방식은 FSM(Finite State Machine)의 상태 할당에 쓰인 방법으로 n 비트의 데이터 워드를 2^n 비트 데이터 워드로 일대일 대응시키는 방법이다. 표 1에 3비트인 경우에 코딩방법을 나타내었다.

표 1. 3비트 데이터의 경우 One-Hot 코딩 테이블

3비트 데이터	One-Hot 코딩	3비트 데이터	One-Hot 코딩
000	00000001	100	00010000
001	00000010	101	00100000
010	00000100	110	01000000
011	00001000	111	10000000

원래의 데이터에 대한 one-hot 코딩 방식의 확률비는 식 (2)와 같다. 만약 n이 8일 경우, switching activity 의 52.8%를 줄일 수 있지만 부가적으로 256개의 라인 이 필요하기 때문에 실제적으로 불가능한 방법이다[5].

$$\frac{P_{one-hot}}{P_{no coding}} = \frac{4(1-2^{-n})}{n} \quad \text{for } n \geq 2 \quad (2)$$

3.2 Gray 코딩

이 방식은 데이터의 패턴이 순차적이거나 데이터들의 연관성이 높을 때 유용하며 부가적인 컨트롤 라인이 필요 없어 어드레스 버스 라인에 적용시킬 수 있다. 식 (3)과 같이 gray 엔코딩 방식은 원래 데이터의 MSB(Most Significant Bit)값을 기준으로 LSB(Least Significant Bit)까지 하위 인접한 2개 비트와의 배타적 논리합(XOR)을 취하여 값을 얻으며, 식 (4)와 같이 코딩된 데이터의 MSB 값을 기준으로 복원된 원래 데이터 비트와 그 하위 코딩된 데이터 비트와의 배타적 논리합을 취하여 원래의 데이터를 복원할 수 있다. 여기서 b_i 는 원래 이진 데이터를 나타내고, g_i 는 gray 코드를 나타내며, 표 2에 이진수 및 gray 코드 값을 나타내었다.

표 2 십진수, 이진수 및 gray 코드

십진수	이진수	gray 코드	십진수	이진수	gray 코드
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

$$g_{n-1} = b_{n-1}$$

$$g_i = b_{i+1} \oplus b_i \quad (i=n-2, n-3, \dots, 1, 0) \quad (3)$$

$$b_{n-1} = g_{n-1}$$

$$b_i = b_{i+1} \oplus g_i \quad (i=n-2, n-3, \dots, 1, 0) \quad (4)$$

예를 들어, 원래의 4비트 데이터 B={1,1,0,1}에 대해

서 gray 코딩식은 $G=\{b_3, b_3 \oplus b_2, b_2 \oplus b_1, b_1 \oplus b_0\}$ 이므로 최종 코딩된 값은 {1,0,1,1}이 된다. 또한, 코딩된 데이터 $G=\{1,1,0,1\}$ 에 대해서, 원래의 4비트 데이터를 복원하기 위한 식은 $B=\{g_3, g_3 \oplus g_2, g_2 \oplus g_1, g_1 \oplus g_0\}$ 이므로, 최종 디코딩된 값은 {1,0,0,1}이 된다.

3.3 Bus-Inversion 코딩

이 방식은 데이터 패턴을 Gaussian 분포를 갖는 임의의 데이터로 가정하고 접근한 방식으로 디코더는 간단한 구조로 구현할 수 있으며, 원래 데이터 라인 이외에 1개의 제어 라인만이 필요하기 때문에 부가적인 비용이 가장 적다.

이 방식은 원래 전송할 데이터를 'data value'로, 코딩된 값을 'bus value'로, 부가적인 제어 라인을 'invert'로 정의하고, 현재의 bus value와 다음 data value사이의 Hamming distance(두개의 데이터 사이의 차이 개수) 값을 계산하여 데이터 라인 수가 n이라고 할 때 n/2보다 큰 경우 invert를 1로 정하고 다음 bus value를 반전하여 다음 data value로 정한다. 반대로 Hamming distance 값이 n/2보다 작거나 같은 경우에는 invert를 0으로 정하고 다음 bus value를 그대로 다음 data value로 정한다. 그러므로, 디코더에서는 invert 값이 1인 경우 쉽게 bus value를 반전시켜 원래 data value를 얻을 수 있다. 예를 들어, 그림 3(a)에 원래의 데이터를 그림 3(b)에 원래의 데이터를 bus-inversion 코딩 방식으로 코딩된 데이터를 도시하였다.

D0	1000010011011000	→	D0	1000000100110101
D1	1000010101011010		D1	1000000010000001
D2	0110010100010011		D2	0110000011111110
D3	1111000011000010		D3	1111010100101111
D4	0001100001110010		D4	0001110110011111
D5	0101010110011001		D5	0101000001110100
D6	11001110000101001		D6	1100101111000100
D7	1100010110010010		D7	1100000001111111
		inv	0000010111101101	

(a) 원래의 데이터

(b) 코딩된 데이터

그림 3 bus-inversion 코딩 방식 예제

이 방식은 식 (5)에서 알 수 있듯이, n 비트의 버스를 나누어 각각의 2 비트 라인에 대해 invert를 할당하는 경우 가장 적게 줄일 수 있지만(25%), 8-bit bus인 경우에는 제어 라인인 invert가 4개 필요하게 되므로 본 논문에서 적용한 LCD 시스템에서는 바람직한 방법은 아니다. 일반적인 8-bit bus에 하나의 invert 라인을 적용할 경우 이론적으로 18.2% switching activity의 감

소를 기대할 수 있다[4].

$$\frac{P_{coding}}{P_{no\ coding}} = \left(1 + \frac{1}{n}\right) \left(1 - \frac{C}{2^n}\right)^{n/2} \quad (5)$$

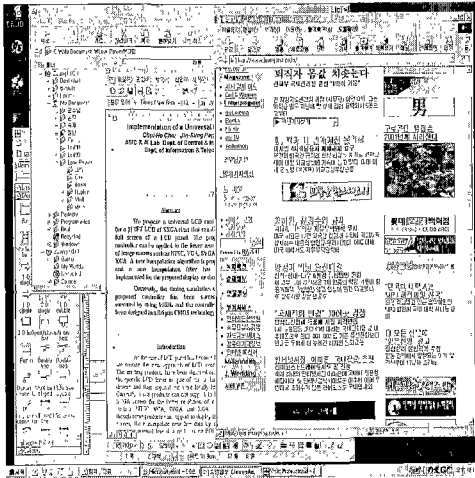
4. 제안한 Signaling 방법

4.1 컴퓨터에서 일반적인 텍스트 화면의 데이터 분포

본 논문에서 제안한 코딩 방법은 LCD 컨트롤러와 구동 칩 사이의 데이터 라인(그림 1)에 적용한 방식으로 먼저 컴퓨터의 일반적인 텍스트 화면의 데이터 분포를 분석할 필요가 있다. 그림 4에 256 계조(8 비트의 R(Red), G(Green), B(Blue) 데이터로 하나의 픽셀을



(c) 샘플 3



(a) 샘플 1

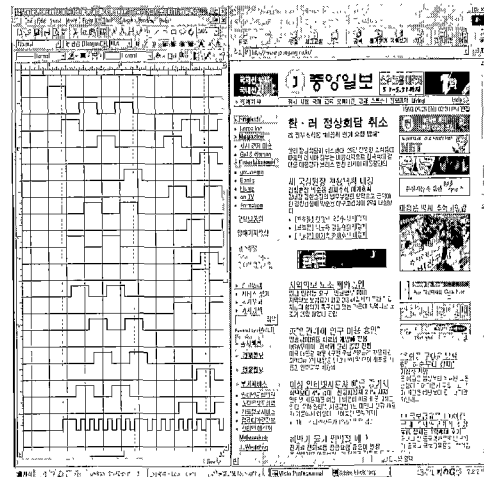


(d) 샘플 4

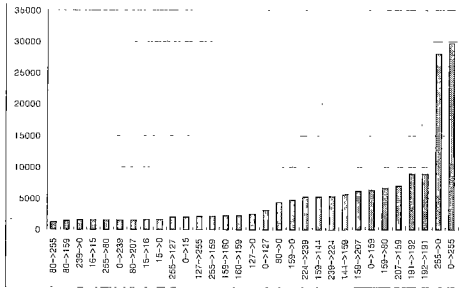
그림 4 일반적인 텍스트 화면과 그림 화면의 샘플

표현)를 갖는 일반적인 텍스트 화면과 그림 화면을 나타내었고, 그림 5에는 각 샘플에 대한 R, G, B 데이터 중 R 데이터의 비트별 천이 분포 중에서 그 개수가 상위 1% 이상인 것만을 나타내었다.

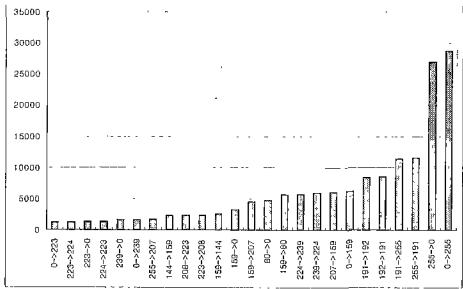
그림 5에서 나타낸 것처럼 텍스트로만 이루어진 화면에 대해서는 명조 대비가 큰 색상으로 텍스트를 표현하기 때문에 밝은 색에서 어두운 색으로 또는 어두운 색에서 밝은 색으로 천이 되는 분포가 가장 많이 차지한다. 반면 그림의 경우에는 색상의 분포가 임의적이고 인접 픽셀 값의 천이는 순차적인 분포를 갖고 있음을 알 수 있다.



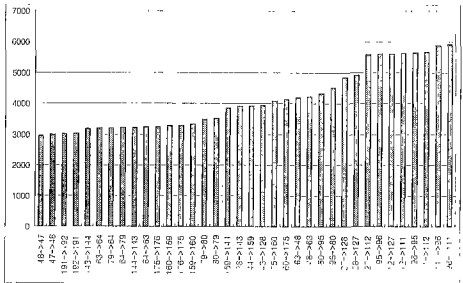
(b) 샘플 2



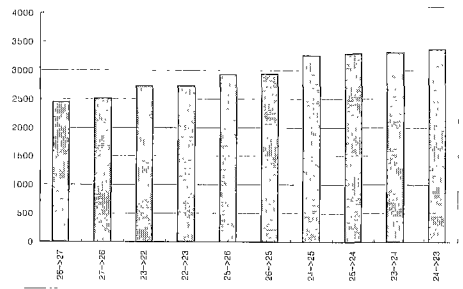
(a) 샘플 1의 데이터 천이 분포



(b) 샘플 2의 데이터 천이 분포



(c) 샘플 3의 데이터 천이 분포



(d) 샘플 4의 데이터 천이 분포

그림 5 각 샘플의 R 데이터 천이 분포

예를 들어, 그림 5(a), (b)의 천이 분포 결과를 살펴 보면, “255₍₁₀₎”(흰색)값에서 “0₍₁₀₎”(검정)값으로, “0₍₁₀₎”에서 “255₍₁₀₎”으로 천이하는 분포가 가장 많이 차지하고

있으며, 또한 이 값들은 각각 이진수로 “00000000₍₂₎”과 “11111111₍₂₎”를 나타내기 때문에 비트의 천이가 가장 많이 발생한다는 것을 알 수 있다. 이는 텍스트 화면 특성상 흰바탕색에 검은색 문자를 표현하고 있기 때문에 이 두가지 색의 분포가 가장 많으며, 조합 또한 불규칙적이기 때문에 비트별 천이가 최대로 발생하기 때문이다.

4.2 제안한 코딩 방법 및 구현

그림 5에서 분석한 텍스트 화면의 데이터 천이 분포를 고려한 엔코딩 및 디코딩 방법을 식 (6) 과 (7)에 각각 나타내었다. 부가적인 제어 신호없이 코딩된 데이터를 원래의 데이터로 복원하고, 입력 값이 “01₍₂₎” 혹은 “10₍₂₎”일 때 “1₍₂₎” 값을 출력하고, “11₍₂₎” 혹은 “00₍₂₎”일 때 “0₍₂₎” 값을 출력하여 천이를 감소시키는 배타적 논리 합을 이용하였다. 그림 5(d)의 그림 화면인 경우 데이터의 순차적인 조합 특성(“26₍₁₀₎→27₍₁₀₎”, “27₍₁₀₎→26₍₁₀₎”, “25₍₁₀₎→26₍₁₀₎”, “26₍₁₀₎→25₍₁₀₎” 등)을 감안하여 데이터의 인접 비트별 배타적 논리 합을 취하여 1차 코딩된 데이터를 얻으며 그림 5(a)(b)(c)의 텍스트 화면인 경우 흰색과 검정색과 같은 명도 대비가 큰 데이터들의 조합(“255₍₁₀₎→0₍₁₀₎”, “0₍₁₀₎→255₍₁₀₎” 등)을 고려하여 현재 데이터와 이전 데이터의 동일 비트별 논리 합을 취하여 최종 코딩된 데이터를 얻는다. 여기서 B는 전송할 원래 데이터를 나타내고, G’은 1차 코딩된 데이터를, G는 실제 전송되는 2차 코딩된 데이터를 의미한다.

$$G'_{n-1} = B_{n-1}$$

$$G'_i = B_{i+1} \oplus B_i \quad (i=n-2, n-3, \dots, 1, 0)$$

$$G_i[t] = G'_i[t] \oplus G'_i[t-1] \quad (i=n-1, n-2, \dots, 1, 0) \quad (6)$$

$$B'_i[t] = G'_i[t] \oplus G'_i[t-1] \quad (i=n-1, n-2, \dots, 1, 0)$$

$$B_{n-1} = G'_{n-1}$$

$$B_i = B'_{i+1} \oplus G'_i \quad (i=n-2, n-3, \dots, 1, 0) \quad (7)$$

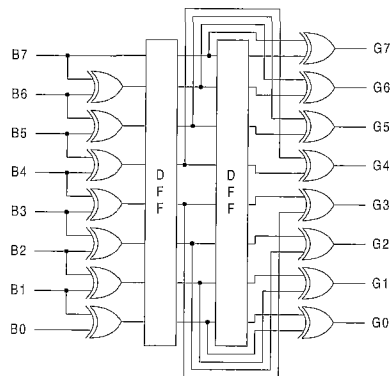


그림 6 제안한 엔코딩 방법의 하드웨어 구조

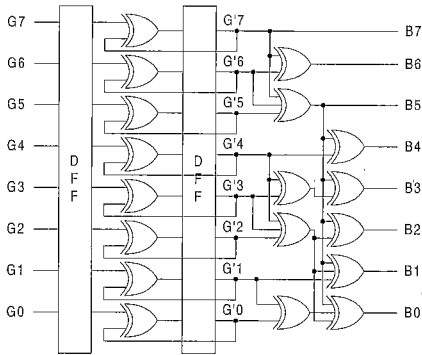


그림 7 제안한 디코딩 방법의 하드웨어 구조

그림 6과 7에 인코딩 및 디코딩 방법의 하드웨어 구조를 각각 도시하였다. 그림 6에서 인접 비트별로 연결된 7개의 XOR 게이트로 1차 코딩된 데이터를 얻으며, DFF(D-type Flip Flop)에 구분되는 현재 데이터와 이전 데이터와의 동일 비트별로 연결된 8개의 XOR 게이트로 최종 코딩된 데이터를 얻는다. 제안한 방식은 8 비트 버스의 경우 15개의 XOR 게이트로 구현 가능하며 파이프라인 구조를 하고 있어 실시간 처리를 요하는 LCD 구동 시스템에 적합하다.

표 3은 기존의 signaling 방식인 bus-inversion 방식과 본 논문에서 제안한 방식의 하드웨어 성능 분석을 위하여 Synopsys사의 Design Compiler를 이용하여 게이트 수, 전력 소비, 지연 시간 등을 비교 분석한 결과를 보여준다. 여기서, 게이트 수는 inverter 회로의 게이트 수를 1로 정했을 때의 상대적인 개수를, 전력 소비는 공급 전압이 3.3V일때의 동적인(dynamic) 전력(net switching power)을, 지연 시간은 입력된 신호가 최종 출력 될 때까지의 시간을, latency는 파이라인 구조에서 입력 신호가 처리되어 올바른 값이 출력되기까지의 소요 클럭 수를 나타낸다. 표 3에서 bus-inversion 방식은 디코더 회로는 간단하지만, 인코더 회로가 제안한 방

표 3 각 방식의 하드웨어 성능 분석

구 분		Bus Invert		제안한 코딩방법	
		Tx	Rx	Tx	Rx
게이트 수	개별	254	24	178	190
	총계	278		368	
전력소비(mW)	개별	17.60	2.93	7.75	6.56
	총계	20.53		14.31	
지연시간(ns)	개별	6.10	0.39	1.81	2.98
	총계	6.49		4.79	
Latency(clock)		3		4	

식에 비해서 하드웨어와 switching activity 측면의 성능이 매우 낮아 전체적 성능이 제안한 방식보다 떨어짐을 보여준다.

4.3 시뮬레이션 결과

그림 4의 4개 샘플 데이터를 이용하여 3장에서 소개한 기존의 방식과 본 논문에서 제안한 방식을 C 언어로 시뮬레이션을 수행하여 비트별 데이터 천이 개수와 감소율을 비교하였고, 그 결과를 표 4와 그림 8에 도시하였다.

표 4 각 방식의 데이터 천이 개수와 감소율

구 분	샘플 1	샘플 2	샘플 3	샘플 4
BYPASS	3123048	72666304	3320455	7834878
	0%	0%	0%	0%
Bus Invert	1619692	1176678	2776395	7383422
	48%	56%	16%	6%
제안한 코딩 방법	1041750	961327	1195874	6166835
	67%	64%	64%	21%

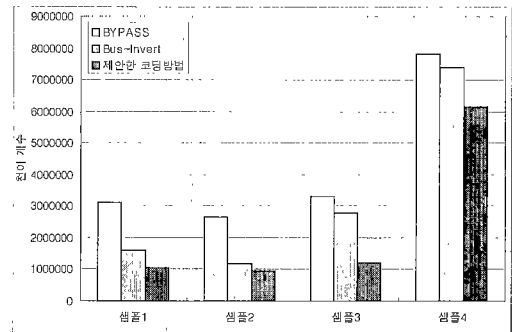


그림 8 각 방식의 성능 비교

그림 8에서 보여주듯이, 본 논문에서 제안한 방식이 컴퓨터의 전형적인 텍스트 화면(샘플 1)에 대해서 switching activity를 최대 67% 줄일 수 있다는 것을 확인할 수 있다. 반면, 그림 화면(샘플 4)에 대해서는 21%의 감소율을 나타내었는데 그 이유는 데이터 천이 분포가 텍스트 화면에 비하여 크지 않으며 데이터 변화 또한 다양한 패턴이 소수로 존재하기 때문이다. 또한 임의의 데이터에 대해서 이론적으로 성능이 우수할 것으로 예상되었던 bus-inversion 코딩은 제안한 방식에 비교하여 텍스트 화면이나 그림 화면에 대해서 좋은 결과를 얻지 못하였다. 왜냐하면 이 코딩 방법은 데이터 패턴을 이상적인 Gaussian 분포에 기반을 두었기 때문이다.

5. 결론

본 논문에서는 최근 차세대 디스플레이 장치로서 각광을 받고 있는 LCD 시스템에서 전력 소비와 EMI를 줄이기 위한 방법으로 LCD 컨트롤러와 구동 칩 사이의 인터페이스에서 switching activity를 효과적으로 줄일 수 있는 병렬 signaling 방식을 제안하였다. 객관적인 성능 평가를 위해서 다양한 샘플 데이터를 이용하여 기존의 signaling 방식과의 데이터 전이 개수를 카운트하여 비교하였다. 제안한 방식은 컴퓨터의 일반적인 텍스트 화면의 데이터 전이 분포를 고려한 방법으로 시뮬레이션 결과 switching activity를 최대 67%까지 줄일 수 있는 것으로 확인되었다.

제안한 방식은 데이터 전이를 감소시키는 방법에 초점을 두고 있으며, 구조가 단순하여 적은 비용의 추가로 하드웨어 구현이 용이하고, 실시간 처리가 용이한 파이프라인 구조로 이루어져 있다. 따라서, 기존의 데이터 코딩없이 데이터를 그대로 미소 전류로 변환하여 전송하던 차동전류구동기(Differential Current-mode Driver)와 병합하여 signaling 회로를 구성한다면 적은 비용의 추가로 전력 소비 및 EMI를 더욱 효과적으로 감소시킬 수 있어 실시간 처리가 요구되는 LCD 구동 시스템에 적절하게 응용될 수 있다.

참 고 문 헌

- [1] J. A. Castellano, "Market & Technology Trends in the Global Flat Panel Display Market," Proc. Asia Display, pp. 259-264, 1998.
- [2] M. Choi et al., "A-shaped Interpolation Algorithm for a-Si TFT LCD Display," Korea Symposium on Information, pp. 25-26, 1998.
- [3] Edgar Sanchez-Sinencio et al., *Low-Voltage/Low-Power Integrated Circuits and Systems*, IEEE Press, 1999.
- [4] M. Stan et al., "Bus-Invert Coding for Low-Power I/O," Trans. of VLSI Systems, pp. 49-58, March 1995.
- [5] A. P. Chandrakasan and R. W. Brodersen, *Low-Power Digital CMOS Design*, Kluwer Academic, 1995.
- [6] C. H. Choi and M. R. Choi, "A New Parallel Signaling Algorithm for Reducing Switching Activity In a LCD Driving System," TENCON'99, pp. 852-855, 1999.



최 철 호

1998년 한양대학교 제어계측공학과 졸업(학사). 2000년 한양대학교 제어계측공학과(공학석사). 2000년 3월 ~ 현재 한양대학교 전자전기제어계측공학과 박사과정. 관심분야는 ASIC, Low-Pwer/Low-Voltage Circuit Design, VLSI,

RF System Design.



최 명 렬

1983년 한양대학교 전자공학과 졸업(학사). 1995년 미시간주립대학교 컴퓨터공학과(공학석사). 1991년 미시간주립대학교 컴퓨터공학과(공학박사). 1991년 3월 ~ 10월 생산기술연구원 전자정보실용화센터 조교수. 1991년 11월 ~ 1992년 8월 생산기술연구원산하 전자부품종합기술연구소 선임 연구원. 1992년 9월 ~ 현재 한양대학교 전자컴퓨터공학부 부교수. 관심분야는 ASIC, 신경회로망 칩 설계, 스마트카드 응용, $\mu P/DSP$ 응용, Wireless ATM, ITS.