

論文96-33B-1-9

MPEG DCT 계수의 특징을 이용한 효율적인 VLC/VLD의 VLSI 설계

(VLSI design of efficient VLC/VLD utilizing the characteristics of MPEG DCT coefficients)

孔 鍾 弼 * , 金 榮 民 *

(Jong Pil Kong and Young Min Kim)

요 약

본 논문은 가변길이코드의 encoding/decoding을 위한 간단하고도 메모리 측면에서 효율적인 구조를 제안한다. MPEG1 DCT계수를 encoding/decoding함으로써 구현한 본 구조에서 114개의 DCT계수를 메모리 매핑하는데 최소인 7비트의 어드레스가 할당되도록 하였고, 직렬-병렬 및 병렬-직렬 변환용 쉬프트 레지스터와 code mapping ROM을 결합시킨 구조로써 최소의 플립플롭 및 메모리를 사용하여 구현하였다. 속도측면에선 COMPASS tool(0.8 μ m CMOS technology standard cells)을 사용해서 시뮬레이션 해본 결과 encoding/decoding의 경우 모두 50Mbps의 동작속도를 얻을 수 있었다.

Abstract

In this paper we propose an architecture for VLC(Variable Length Coder) and VLD(Variable Length Decoder) which is simple with respect to implementation point and efficient in memory. We implemented encoding and decoding circuit where we need only 7-bit address memory space for 114 MPEG1 DCT coefficients and employed minimal number of flip-flops and logics for an architecture to integrate a shift register for serial-to-parallel or parallel-to-serial conversion of the data in code mapping ROM. We obtained 50Mbps operating speed in both encoding and decoding process as the result of simulation using 0.8 μ m CMOS standard cells.

I. 서 론

영상 데이터를 압축하는 목적은 원래의 영상이 전달하고자 하는 의미를 손상시키지 않는 범위 내에서 가능한 한 적은 양의 정보로 표현하는 데 있다. 특히 최근의 고화질 화상의 저장, 복원 및 전송에 대한 관심은 이들 압축기술을 실제로 구현하는데 큰 동기를 부여하고 있다^[1]. 최근의 영상 데이터 압축의 경우에는 CCITT H.261, MPEG에 나타난 바와 같이 한 영상 내의 중복성과 영상과 영상사이의 중복성 그리고 색상

보다는 휘도 영상신호에 민감한 사람의 인식 기능 등을 이용하여 실제 영상에서의 화질 저하 없이도 상당량의 데이터를 감축하기 위해 손실부호화 방법을 사용한 다음 비손실부호화 방법을 따르게 하여, 결국 두가지 압축방법을 병행하여 사용하고 있다^[2,3,4]. 대표적인 비손실 부호화 방법의 하나인 가변장 부호(variable length code)는 발생빈도가 높은 심볼에 대해서는 길이가 짧은 코드를 그리고 발생빈도가 낮은 심볼에 대해서는 길이가 긴 코드를 할당하여 부호화 효율(coding efficiency)을 높이는데 사용되는 코드로 더 적은 양의 평균 비트를 사용하면서도 같은 양의 정보를 표현하는 것이 가능하다^[5,6,7].

VLC 복호기를 구현하는 방법에는 tree search 방

* 正會員, 全南大學校 電子工學科

(Dept. of Elec. Eng., chonnam National Univ.)

接受日字: 1995年2月17日, 수정완료일: 1995年12月19日

식이 사용되는 비트직렬(bits serial) 구조와 병렬정합(parallel match) 방식이 사용되는 비트병렬(bit parallel) 구조가 있다^[8]. 비트직렬 구조는 일반적으로 한 클럭에 한 심볼을 처리하는 비트병렬 구조에 비해 비트처리 속도에서 느리지만 구현이 쉽고 단순한 구조를 가질 수 있으며, 이 단순구조에 의해, 상대적으로 많은 블록이 요구되고 블록들간의 동기화 또는 더 커진 블록자체의 지연이 요구될 수 있는 병렬구조에 비해 클럭 속도의 증가가 용이하여 최적화 구조의 하드웨어를 구현할 경우 비트병렬 구조에 상응하는 비트처리 능력을 가질 수 있다. 이 두 구조는 입력되는 압축 비트 데이터의 전송율에 따라 연산 속도를 달리한다^[9-11].

본 논문에서는 MPEG1에서 규정된 VLC 코드를 효과적으로 부호/복호하기 위한 단순한 구조와, 메모리 측면에서 효과적인 어드레싱 방법을 제시하고 이를 직접 설계하여 그 결과를 확인함으로써 최근 국제규격으로 공인된 MPEG2의 한정된 규격에 적용 가능성을 보인다. 본 논문의 구성은 다음과 같다. II장에서는 가변길이코드의 복호화기 설계, III장에서는 가변길이코드의 부호화기의 설계를 그리고 마지막으로 IV장에서는 결론을 맺었다.

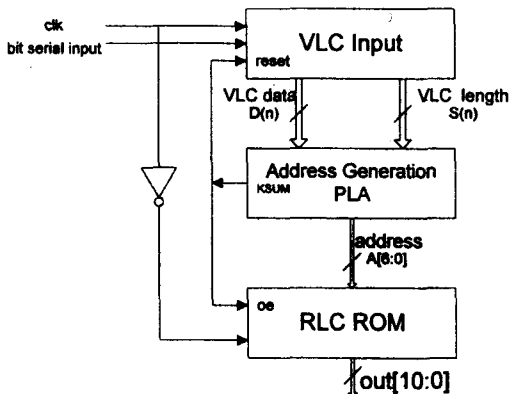


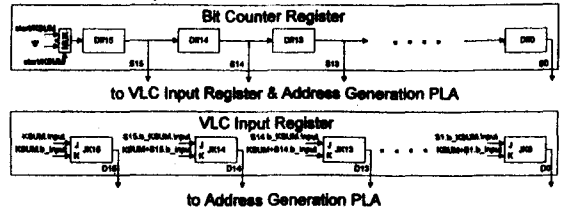
그림 1. 비트직렬(bit serial) 가변길이코드 복호기 블록도

Fig. 1. Block diagram of bit serial VLC decoder.

II. 가변길이 코드 복호기

1.5Mbps 정도의 압축 비트 데이터 전송율을 가지는 MPEG1 시스템에서 tree search 구조의 경우에는 입력되는 압축 비트 데이터 전송 속도와 같은 1.5MHz

의 연산 속도를 필요로 하는바 본 논문에서는 단순한 구조의 비트 직렬 복호기를 설계한다. 이는 그림 1에 나타난 바와 같이 크게 VLC Input 블록, Address Generation PLA 블록, RLC(Run Length Code) ROM 블록의 세 블록으로 나눌 수 있다.



* 각 신호 사이의 “.”연산은 논리곱, “+”연산은 논리합, “b_”는 부논리를 의미

그림 2. 가변길이부호 입력 블록도

Fig. 2. VLC input block diagram.

1. VLC(variable length code) Input 블록

이 블록은 그림 2에 나타난 바와 같이 입력되는 가변길이부호 비트를 카운트하는 16개의 D플립플롭으로 구성된 Bit Counter Register와 입력되는 가변길이부호 비트를 좌측의 MSB 플립플롭부터 차례로 받아들이는 16개의 JK플립플롭으로 구성된 VLC Input Register로 구성된다.

Bit Counter Register는 start 신호나, Address Generation PLA 블록의 KSUM 신호가 '1'이 된 후 다음의 첫번째 클럭에 의해 MSB 플립플롭인 Dff15가 '1'이 되고, 매 클럭마다 Dff14, Dff13, ..., Dff0으로 전파되어 임의의 심볼에 해당하는 코드의 모든 비트가 입력될 때까지 이 '1'이 쉬프트된다. MPEG1 DCT 계수의 가변길이부호의 최대길이가 16비트에 근거해서 D, JK 플립플롭의 수를 각각 16개로 구성한 그림 2를 예로 들면 다음과 같다. 만약 한 심볼이 4비트의 길이를 가진 코드라면 첫번째 클럭에 Bit Counter Register의 S15, 두번째 클럭에 S14, 세번째 클럭에 S13이 '1'이 되어 다음 4번째 클럭에는 S12가 '1'이 되고 결국 VLC Input Register의 D12에 VLC code의 4번째 비트가 입력되어 심볼정합이 완료된다. 결국 Bit Counter Register는 '1'을 최상위 플립플롭인 D15부터 시작하여 하위 비트 쪽으로 쉬프트 시킴으로써 입력되는 VLC 데이터를 카운트하면서 임의의 심볼에 해당하는 VLC 비트 데이터가 VLC Input Register의 적정 플립플롭에 입력되도록 위치를 정해

주는 역할을 한다.

VLC Input Register는 전술된 바와 같이 임의의 한 심볼에 해당하는 VLC 비트 데이터를 MSB 플립플롭부터 직렬로 받아들이고 다시 Address Generation PLA 블록의 KSUM 신호에 의해 초기화하는 과정을 되풀이한다. 이때 KSUM, Sn, input(VLC 입력 데이터)을 입력으로 하는 각 JK 플립플롭들의 입력은 그림 2의 VLC Input Register로부터 볼 수 있듯이 다음 식 1과 같다.

$$\begin{aligned}
 J15 &= KSUM \cdot input \\
 K15 &= KSUM \cdot b_input \\
 J14 &= S_{15} \cdot b_KSUM \cdot input \\
 K14 &= KSUM + S_{15} \cdot b_input \\
 J13 &= S_{14} \cdot b_KSUM \cdot input \\
 K13 &= KSUM + S_{14} \cdot b_input \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 J0 &= S_1 \cdot b_KSUM \cdot input \\
 K0 &= KSUM + S_1 \cdot b_input
 \end{aligned}$$

식 1.

여기서 KSUM은 VLC 코드 정합 신호, Sn은 Bit Counter Register의 출력, input은 입력되는 VLC 비트 데이터를 그리고 "b_"는 부논리를 의미한다.

다음의 표 1은 VLC 입력으로 7비트 "0000 110"과 3비트 "011" 두개의 심볼이 입력되었을 때 Bit Counter Register와 VLC Input 블록에 있는 각 플립플롭의 출력인 Sn과 Dn 및 정합발생 신호인 KSUM의 상태를 나타낸다. 표 1에서 D플립플롭으로 구성된 Bit Counter Register의 최상위 비트(MSB)인 S15가 '1'로 되는 첫번째 클럭에서는 VLC의 디코딩이 시작되고, 8번째 클럭에서는 VLC의 정합이후 새로운 VLC의 디코딩이 시작됨을 나타낸다. 그리고 KSUM은 전술된 바와 같이 '1'로 되는 7번째 클럭과 10번째 클럭에서 VLC가 정합 되었음을 나타내고, 다음 클럭인 8번째, 11번째 클럭에서 새로운 코드에 대한 디코딩이 시작됨을 의미한다. VLC Input 블록의 초기화는 start나 KSUM이 '1'로 되는 시점에서 발생한다. 초기화가 되면 Bit Counter Register의 Dff15는 '1'을, VLC Input Register의 JK15는 입력되는 VLC 코드

의 첫 비트를 받아들이고 블록내의 모든 나머지 플립 플롭들은 '0'으로 리셋(reset)된다. 그리고 다음의 그림 3은 Bit Counter Register와 VLC Input Register로 구성된 VLC Input 블록의 로직 다이어그램을 보여준다.

표 1. VLC Input Register의 입출력 상태 변화표
Table 1. Input/Output state transition table of VLC Input Register.

클럭	1	2	3	4	5	6	7	8	9	10
VLC Input	0	0	0	0	1	1	0	0	1	1
KSUM	0	0	0	0	0	0	1	0	0	1
Sn/Dn	S15/D15	1/0						1/0		
	S14/D14		1/0						1/1	
	S13/D13			1/0						1/1
	S12/D12				1/0					
	S11/D11					1/1				
	S10/D10						1/1			
	S9/D9							1/0		
	S8/D8									
	S7/D7									
	S6/D6									
	S5/D5									
S4/D4										
S3/D3										
S2/D2										
S1/D1										
S0/D0										

* 공란은 Sn/Dn이 "0/0"임을 표시

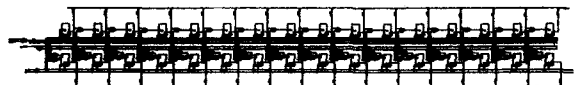


그림 3. VLC Input block의 로직 다이어그램
Fig. 3. Logic diagram of VLC Input block.

2. Address Generation PLA 블록 및 RLC(Run Length Code) ROM 블록

Address Generation PLA 블록은 심볼정합이 발생할 경우 VLC Input 블록의 초기화 신호(KSUM) 및 RLC ROM 블록의 어드레스를 발생하고, RLC ROM 블록은 심볼정합이 발생한 후 출력가능신호(oe)가 '1'인 상태에서 ROM access 시간이 경과한 후에 복호화된 결과를 출력하게 되는데 출력가능신호(oe)는 KSUM신호가 '1'인 상태에서 '1'이 된다.

표 2. 복호화 경우 MPEG1 DCT 계수에 대한 어드레스 생성 PLA로직

Table 2. Address Generation PLA logic for MPEG1 DCT coefficients in decoding process.

kn = VLC 입력	A6 A5 A4	A3 A2 A1 A0
k0 = D15S15	0 0 0	0 1 0 0
k1 = D15D15S14		0 0 0 0
k2 = D15D14D15S13		0 0 0 1
k3 = D15D14D15S12		0 0 0 D12
k4 = D15D14D15S11		0 1 D12 D11
k5 = D15D14D13D12D11S10		1 D10 D9 D8
k6 = D15D14D13D12S10	0 0 1	0 0 D11 D10
k7 = D15D14D13D12D11S9		0 1 D10 D9
k8 = D15D14D13D12D11D10D9S8		1 D8 D7 D6
k9 = D15D14D13D12D11D10D9D8S7	0 1 0	D7 D6 D5 D4
k10 = D15D14D13D12D11D10D9D8D7S6	0 1 1	D6 D5 D4 D3
k11 = D15D14D13D12D11D10D9D8D7D6S5	1 0 0	D5 D4 D3 D2
k12 = D15D14D13D12D11D10D9D8D7D6D5S4	1 0 1	D4 D3 D2 D1
k13 = D15D14D13D12D11D10D9D8D7D6D5D4S3	1 1 0	D3 D2 D1 D0
KSUM = k1+k2+ +k13		

- * 부분의 데이터는 로직 '1' 의미
- * S15,D15는 각각 Bit Counter Register와 VLC Input Register의 MSB

MPEG1의 경우 DCT 계수는 114개의 VLC코드에 의해 표현되는 한정된 범위내의 run/level값과 이 범위를 벗어나는, Escape코드("000001")와 이어지는 고정길이 코드를 더해 20비트 혹은 28비트의 고정길이 코드로 표현되는 run/level값으로 구성된다. VLC코드로 표현되는 MPEG1 DCT 계수에 대한 Address Generation PLA 블럭의 PLA 논리 및 최적 어드레스 할당내용을 예로 보면 표 2와 같은데 이 표 2로부터 114개의 DCT 계수 가변길이부호에 대해 최소 어드레스인 2⁷의 메모리 양이 필요함을 볼 수 있다. 이는 MPEG1 AC계수를 사용하여 새로운 구조를 제안한 참고문헌 [12]의 내용과 비교할 수 있는데, 비록 병렬처리란 점에서 직접적인 비교는 어렵지만 [12]에서 비교대상으로 보인 기존의 방법을 사용했을 때의 2¹⁶, 그리고 [12]에서 제안하는 구조에 의한 (2⁸+

2⁷+2⁷)의 메모리양에 비해 512:4:1로 메모리양이 감소됨을 알 수 있다. 그리고 정합발생 신호인 KSUM은 KSUM=k1+k2+ + k13으로 임의의 한 심볼이라도 정합이 일어날 경우 '1'로 되고, 각 kn은 VLC Input블럭의 Dn과 Sn에 의해 로직이 구성되어 있음을 볼 수 있다.

예를 들면 k7은 D15D14D13D12D11S9로 S9에 의해 7 비트(D15~D9)가 입력되고 D15D14D13D12D11에 의해 상위 5비트는 "00001"이 되어야 '1'로 되고, K6은 D15D14D13D12S10으로 상위 4비트가 "0001"로 시작하고 S10에 의해 모두 6비트(D15D14D13D12D11D10)가 입력되어야 '1'이 됨을 알 수 있다.

ROM으로 구성된 RLC ROM 블럭은 정합 Address generation PLA 블럭으로부터 어드레스를 입력으로 받아 정합신호인 KSUM이 발생하고 ROM access 시간이 경과한 후에 11비트(run value:상위 5비트, level value:하위 6비트)의 출력을 발생하도록 되어 있다.

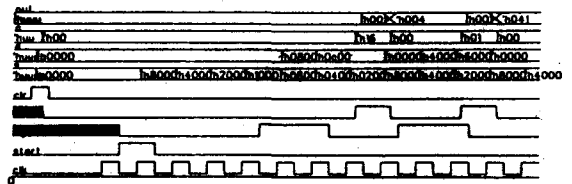


그림 4. 가변길이 복호화기의 COMPASS 시물레이션 결과
Fig. 4. COMPASS simulation result of VLC decoder.

3. 시물레이션 결과

그림 4에 보인 VLSI TECH사의 VLSI tool인 COMPASS(0.8μm CMOS technology standard cell, Vdd=4.65V, Vss=0.1V, T=70 degree C, WORST CASE SPEED)를 사용하여 수행한 시물레이션 결과로부터 표 2의 가변길이부호 입력과 그때의 어드레싱 및 출력을 비교 확인할 수 있다. 그림에서 start 신호가 '1'이 되는 2번째 클럭에서 VLC 디코딩이 시작되어 7비트를 입력받은 8번째 클럭에서 정합신호인 KSUM이 '1'로 되고 이후 3비트를 입력받고 또 정합 신호인 KSUM이 발생되는데 이때 RLC ROM 블럭의 출력 신호인 out [10:0]은 KSUM이 '1'이 된 후 ROM의 access 타임이 지난 후 출력됨을 볼 수 있다. 즉 VLC 입력이 "0000 110"일때 K7 =D15D14

$D_{13}D_{12}D_{11}S_9$ 에 의해 상위 5비트가 "00001"이고 전체 7비트가 입력될 때 k7이 '1'로 된다는 조건에 만족되어 KSUM이 '1'이 된다. 이 신호는 VLC Input 블록으로 피드백(feedback)되어 초기화 신호로 작용하고 새로운 VLC 데이터 입력의 시작점이 된다. 마찬가지로 두 번째 입력인 "011"에 대해선 k2가 '1'이 되어 KSUM을 통해 피드백됨을 알 수 있다. 11비트로 이뤄진 출력신호 out [10:0]은 unsigned value로 복호화된 run과 level값을 나타내는데 전술한 바와 같이 상위 5비트는 run, 하위 6비트는 level값을 나타낸다.

그림 4에 나타난 시뮬레이션 결과에 대한 입력, 이 때의 어드레스 및 출력을 요약하면 다음 표 3과 같다.

표 3. 복호화기 시뮬레이션 결과
Table 3. Simulation result of decoder.

VLC 입력	어드레스	출 력	run / level
0000 110	h'16 (b'001 0110)	h'004(b'00000 000100)	0 / 4
011	h'01 (b'000 0001)	h'041(b'00001 000001)	1 / 1

이 결과는 표 2의 어드레싱과 일치함을 알 수 있으며 이때의 복호화된 출력 역시 입력되는 VLC 코드에 상응하는 run/level 값이 발생함을 알 수 있다. 그리고 속도와 하드웨어 크기 측면에선 50MHz(20ns)의 클럭속도, 620개 정도의 gate가 사용되었다. 이 gate 수는 JPEG표준에 적용하여 5000개의 gates를 사용하여 하드웨어를 구현한 참고문헌 [13]의 구조에 비해서도 상당히 적은양의 하드웨어가 사용되었음을 알 수 있다.

III. 가변길이 코드 부호기

이 장에서는 단순한 구조의 가변길이부호의 부호화기를 MPEG1 DCT 계수의 경우를 예로 들어 구현하는 것으로 그림 5에 보인바와 같이 RLC Input블럭, 해당 가변길이부호가 코딩되어 있는 VLC Coded ROM블럭, 그리고 VLC Output블럭의 세 블럭으로 나눌 수 있다.

1. RLC Input블럭

이 블럭은 5비트의 run값과 6비트의 level값을 입력으로 받아들여 어드레스를 출력하는 블럭으로써 표 4의 로직에 의한 PLA로 구성된다.

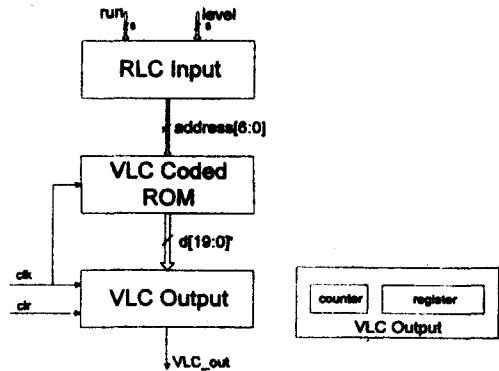


그림 5. 가변길이코드 부호기 블럭도
Fig. 5. Block diagram of VLC encoder.

표 4. 부호화 경우 MPEG1 DCT 계수에 대한 어드레스 생성 PLA로직

Table 4. Address generation PLA logic for MPEG1 DCT coefficients in encoding process.

구 분	A6 A5	A4 A3 A2 A1 A0	run / level
$k1 = R_4R_3R_2R_1R_0$	$k1L_6$	0 0	$L_4 L_3 L_2 L_1 L_0$ R:0, L:1 ~ 31
	$k1L_6$	0 1	$0 L_6 L_2 L_1 L_0$ R:0, L:32 ~ 40
$k2 = R_4R_3R_2R_1R_0L_4$	0 1	$1 L_3 L_2 L_1 L_0$	R:1, L:1 ~ 15
$k3 = L_4L_3L_2L_1L_0K_1K_2$	1 0	$1 0 0 R_1 R_0$	L:5, R:2 #참조
$k4 = L_4L_3L_2L_1L_0K_1K_2$	1 0	$1 0 1 R_1 R_0$	L:4, R:2 ~ 3
$k5 = L_4L_3L_2L_1L_0K_1K_2$	1 0	$1 0 R_2 R_1 R_0$	L:3, R:2 ~ 6
$k6 = L_4L_3L_2L_1L_0K_1K_2$	1 0	$1 R_3 R_2 R_1 R_0$	L:2, R:2 ~ 16
$k7 = L_4L_3L_2L_1L_0K_1K_2$	1 1	$R_4 R_3 R_2 R_1 R_0$	L:1, R:2 ~ 31

R=1, L=16,17,18인 경우의 어드레싱

$k216 = k2L_6L_4L_3L_2L_1L_0$	A:1010000	R:1, L:16
$k217 = k2L_6L_4L_3L_2L_1L_0$	A:1010001	R:1, L:17
$k218 = k2L_6L_4L_3L_2L_1L_0$	A:1010011	R:1, L:18

* ... 부분의 데이터는 로직'1' 의미

* R: run 값, L: level 값

어드레스는 표 4에 나타난 바와 같이 114개의 run/level값을 어드레싱하는데, 최소 비트인 7비트로 최적화 매핑방법이 사용된 것을 특징으로 한다. 이는 단순히 입력되는 5비트의 run값과 6비트의 level값에 의해 11비트로 어드레스를 할당하는 대신 DCT 계수에 대한 run/level의 특징을 이용한 것으로 단순하게

어드레싱 하는 것에 비해 16:1($2^{11}:2^7$)로 메모리양이 감축된다. 즉 이는 작은 run값에 대해선 level값의 변화가 크고, run값이 커지면 level값이 변화가 작아지는 MPEG, CCITT H.261 등에 사용된 run/level로 구성된 2차원 VLC 코드의 특징을 이용한 최적화 방법이라 할 수 있다. 즉 run=0/level=1~40처럼 level 값이 변화하는 경우는 40개의 level값을 변수로 하는 64개의 어드레스에 해당하는 메모리 공간에, run =1/level=1~18의 경우는 어드레스 0110001~0111111과 run=2/level=5를 할당하고 남은 어드레스 1010000,1010001,1010011에 할당한다. 그리고 run 값이 커져 level값의 변화가 없는 경우에 대해서는 오히려 고정된 level값에 대해 run값을 변수로 하는 어드레스 매핑방법을 적용하는데 이는 표 4의 level =1/run=2~31, level=2/run=2~16 등의 경우에서 볼 수 있다. 매핑결과 어드레스 할당이 되지 않은 17개의 어드레스 공간에 대해서는 h'00값이 할당되어 있다.

2. VLC Coded ROM 블록

이 블록은 RLC Input 블록으로부터 7비트의 어드레스를 입력으로 받아 해당하는 가변길이부호와 부호 길이에 대한 정보를 출력시키는 블록으로 MPEG1의 DCT 계수의 경우 출력비트는 가변길이부호에 해당하는 상위 16비트와 출력할 가변길이부호를 카운트하는 4비트, 총 20비트의 출력을 발생한다. 예로 가변길이 부호가 "00101" 5비트인 run/level(0/3) 값을 부호화 하는 경우 출력 20비트는 "0010 1000 0000 0000 0100"으로 코딩된다. 그중 하위 4비트 "0100"은 출력할 가변길이부호 5비트보다 1비트 작은 정수 4를 나타내며 카운터에 로드됨으로써 5비트를 출력하도록 되어 있고, 나머지 16비트는 VLC Output 블록으로 로드 되어 Counter에 의해 해당 비트만큼 MSB부터 출력 하게 된다.

3. VLC Output블록

이 블록은 VLC Coded ROM블록에서 설명되고 그림 6및 그림 7에 나타난 바와 같이 Counter와 Register로 구성된 블록으로 ROM 블록의 상위 16비트는 16개의 플립플롭에 병렬로 로드되고 하위 4비트는 Counter에 로드되어 Counter에 로드된 수보다 1비트 많은 수의 가변길이부호를 1클럭에 1비트씩 출력하도록 되어있다. 레지스터내의 각 플립플롭들은 load

신호가 '1'일때 혹은 카운터의 출력 신호인 q [3:0]이 영(zero)일때만 VLC Coded ROM 블록으로부터 입력되는 VLC code를 병렬로 로드하고, 그 외 경우에 대해선 로드된 값을 한 클럭에 한 비트씩 직렬로 출력하도록 구성된다. 즉 ROM 출력이 h'28004(2진수 b'0010 1000 0000 0000 0100)인 경우 load 신호가 '1'로 되면 b'0100은 카운터로 입력되고 "b'0010 1000 0000 0000"은 MSB부터 그림 6의 맨 오른쪽 플립플롭부터 로드된다. 이후 클럭이 변할때마다 Register는 Counter의 출력이 영(zero)으로 되어 load이 '1'이 될 때까지 MSB부터 해당 VLC code를 출력한다.

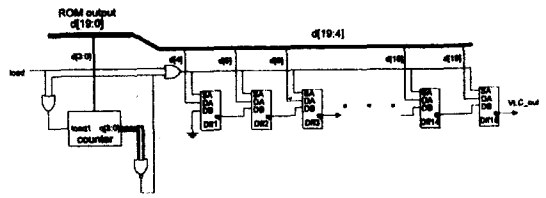


그림 6. 가변길이코드 부호화기의 VLC Output 블록도

Fig. 6. VLC Output block diagram of VLC encoder.

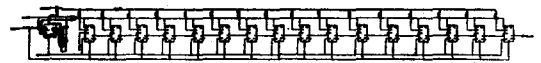


그림 7. 가변길이코드 부호화기의 VLC Output 로직 다이어그램

Fig. 7. Logic diagram of VLC Output block for VLC encoder.

위 그림 6은 가변길이코드 부호화기의 VLC Output 블록도를 그리고 그림 7은 그에 대한 로직 다이어그램을 보여준다.



그림 8. 가변길이 부호화기의 COMPASS 시뮬레이션 결과

Fig. 8. COMPASS simulation result of VLC encoder.

4. 시뮬레이션 결과

그림 8의 VLSI TECH사의 VLSI tool인 COMPASS(0.8 μm CMOS technology standard cell

Vdd=4.65V, Vss=0.1V, T=70 degreè C, WOR-ST CASE SPEED)를 사용하여 수행한 시뮬레이션 결과로부터 표4의 가변길이부호 입력과 그때의 어드레스 및 출력을 비교 확인할 수 있다. 결과에 의하면 VLC Input 블록으로 run=0/level=1이 입력된 후 PLA 로직에 의해 VLC Coded ROM 블록의 어드레스인 a[6:0]="h01"이 출력신호로 발생되고, 그 다음 첫 클럭 이후 ROM access 타임이 경과한 후 VLC Coded ROM 블록으로부터 출력신호인 d[19:0]="hc0001('b 1100 0000 0000 0000 0001)"이 발생한다. 이 출력중 d[3:0]은 load 신호를 받아 VLC Output 블록의 Counter로 로드되고 d[19:4]는 VLC Output 블록의 레지스터로 로드된다. 이때 d[19:4]는 상위 비트가 그림 6에 나타난 바와 같이 오른쪽 플립플롭 Dff15쪽에 로드되는데 MSB는 로드 신호를 받은 두번째 클럭부터 VLC_out으로 출력된다. 이후 카운터 출력 q[3:0]이 영(zero)이 될때까지 레지스터에 로드된 VLC code "11"을 직렬로 출력하고 있다. 두번째 입력 run=1/level=1에 대해선 load1 신호가 '1'이 된후 최종 출력인 VLC_out이 4번째 클럭부터 "011"로 출력됨을 볼 수 있다. 결과로 나타난 어드레스 및 출력상태를 요약하면 다음 표 5와 같다. 그리고 시뮬레이션 결과 클럭 속도는 50MHz(20ns), gate수는 PLA, ROM을 포함하여 380개 정도였다.

표 5. 부호화기 시뮬레이션 결과
Table 5. Simulation result of encoder.

run/level 입력	어드레스	ROM 출력	VLC_out
0 / 1	h'01 (b'000 0001)	b'1100 0000 0000 0000 0001	11
1 / 1	h'31 (b'011 0001)	b'0110 0000 0000 0000 0010	011
0 / 2	h'02 (b'000 0010)	b'0100 0000 0000 0000 0011	0100

이 결과와 표 4를 비교함으로써 각각의 run/level 입력에 대한 어드레스가 일치함을 알 수 있고, 그에 따른 MPEG1 DCT 계수값이 VLC_out으로 올바르게 발생함을 볼 수 있다.

부호화기는 실시간 부호화를 위해선 run/level 입력 블록과 ROM 블록만으로 구성하고 가변길이부호 블록은 버퍼와 결합하여 별도로 구성하는게 바람직하나 본 시뮬레이션에서는 그 출력값을 확인하기 위한 수단으로 ROM 블록위에 바로 가변길이 출력블록을 추가하

여 구현하였다.

IV. 결 론

MPEG1의 DCT계수에 대한 가변길이부호에 대해서 16개의 플립플롭으로 구성된 입력블록을 사용하는 간단한 정합 구조를 갖는 부호화기와 메모리 측면에서 효율적인 부호화기를 설계하는데 있어서의 방법 및 시뮬레이션 결과를 기술하였다. 부호화기는 간단한 입력 블록과 PLA에 의해 설계 되었으며, 비교된 기존의 방법에 비해 메모리양에 있어서 4배이상 감소하였고, 부호화기의 경우 MPEG1의 run/level에 대한 VLC코드의 특징을 이용하여 어드레싱을 한 결과 단순하게 어드레싱을 하는것에 비해 16배정도의 메모리양을 줄일 수 있었다.

본 논문에서 제안된 방법은 MPEG1, MPEG2, CCITT H.261 등의 cbp.mb_address_increment 등의 가변길이부호를 사용하는 다른 매개변수에 대해서도 동일한 방법으로 적용할 수 있을 뿐 아니라 속도 측면에서도 VLSI TECH사의 VLSI tool인 COMPASS(0.8um CMOS technology standard cell)를 사용해서 시뮬레이션 한 결과 decoding/encoding시의 클럭 속도가 각각 50MHz/50MHz로 MPEG2의 MP@ML등의 한정된 규격의 시스템에 대한 지원이 가능하다.

참 고 문 헌

- [1] 김영민외 5명, 영상통신기용 VSP 프로세서의 VLSI 개발에 관한 연구, 상공자원부 중간보고서, 1994년
- [2] "Video codec for audio visual services at px64 Kbits/s." CCITT Recommendation H.261, 1990.
- [3] Motion Picture Expert Group, "Doc. ISO-IEC 11172-2", 1993.
- [4] Motion Picture Expert Group, "Doc. ISO-IEC 13818-2", 1994.
- [5] R.W Hamming, "Coding and Information Theory," PRENTICE-HALL, 1986.
- [6] A.Mukherjee, N.Ranganathan, and M. Bassiouni, "Efficient VLSI design for

- data transformation of tree-based codes," IEEE Trans. on Circuits and System, pp. 306-314, 1991.
- [7] Byeungwoo Jeon, Juha Park, and Je-chang Jeong, "Application of Dynamic Huffman Coding to Image Sequence Compression," SPIE vol. 2308, pp. 1636-1647.
- [8] 황기수의 2명, "멀티미디어 컴퓨터 핵심 VLSI 기술," 전자공학회지 Vol. 20, No. 11, pp. 27-35, 1993년 11월
- [9] Hashemian. R. "Design and hardware implementation of a memory efficient Huffman decoding," IEEE Trans. on Consumer Electronics, pp. 345-352, 1994.
- [10] Heonchul Park and Viktor Prasanna, "Area Efficient VLSI Architecture for Huffman Coding," IEEE Trans. on Circuit and System-II. Analog and Digital Signal Processing, vol. 40, No. 9, pp. 568-575, sep. 1993.
- [11] S.Y. Pyon, S.H. Nam, Y.S. Cheon, K.S. Sohn, "A Programmable VLSI Architecture for Variable Length Decoder," JTC-CSCC'94 vol II, pp. 944-949, JULY, 1994.
- [12] 김이한, 김성대, 이상지, "코드분류에 의한 병렬처리 VLC 복호," 제5회 신호처리합동학술대회 논문집, 제5권, 제1호, pp. 520-523, 1992년 9월
- [13] 김병한, 임종석, "한쪽으로 기운 허프만 트리에서 효율적인 허프만 복호 기법," 한국통신학회 논문지'93-12 Vol. 18 No. 12, pp 1596-1969

 저 자 소 개



金榮民(正會員)

1954년 4월 18일생. 1976년 2월 서울대학교 전자공학과 졸업(공학사). 1978년 2월 한국과학기술원 전기및전자공학과 졸업(공학석사). 1978년 3월 ~ 1979년 7월 한국선박해양 연구소(주임연구원). 1979년 8월 ~ 1982년 7월 국방과학연구소(연구원). 1986년 오하이오 주립대학교 전기공학과(공학박사). 1988년 6월 ~ 1991년 8월 한국전자통신연구소(실장). 1991년 9월 ~ 현재 전남대학교 전자공학과 교수. 주요관심분야는 영상처리, VLSI 설계, 신경회로망 등임.



孔鍾弼(正會員)

1966년 3월 3일생. 1991년 2월 전남대학교 전자공학과 졸업(공학사). 1991년 1월 ~ 1993년 12월 대우전자(주). 1994년 3월 ~ 현재 전남대학교 전자공학과 대학원 재학중. 주요관심분야는 VLSI 설계, 영상압축 등임.