
목시적 가중 예측기법을 이용한 저 메모리 대역폭 인터 예측기 설계

김진영* · 류광기**

Design of a Low Memory Bandwidth Inter Predictor
Using Implicit Weighted Prediction Technique

Jinyoung Kim* · Kwangki Ryoo**

본 논문은 교육과학부와 한국연구재단의 지역혁신인력양성사업 및 지식경제부 출연금으로
수행한 ETRI SW-SoC 융합 R&BD 센터와의 공동 연구의 결과입니다.

요 약

본 논문에서는 H.264/AVC 인코더의 성능 향상을 위해 다중 참조 프레임 기법과 목시적 가중 예측 기법을 이용하
고 낮은 외부 메모리 접근율을 위해 이전 참조 프레임 데이터를 재사용하는 인터 예측기 하드웨어 구조를 제안한
다. 참조 소프트웨어 JM16.0과 비교하여 참조 프레임 접근율이 약 24%만큼 감소하고 참조 영역 메모리가 약 46%만
큼 감소하였다. 통합 구조는 Verilog HDL로 설계되고 Magnachip 0.18um 공정으로 합성한 결과 게이트 수는 약
2,061k 이고 91Mhz로 동작한다.

ABSTRACT

In this paper, for improving the H.264/AVC hardware performance, we propose an inter predictor hardware design using a multi reference
frame selector and an implicit weighted predictor. previous reference frame are reused for Low Memory Bandwidth. The size of the reference
memory in the predictor was reduced by about 46% and the external memory access rate was reduced by about 24% compared with the one in
the reference software JM16.0. We designed the proposed system with Verilog-HDL and synthesized inter predictor circuit using the
Magnachip 0.18um CMOS standard cell library. The synthesis result shows that the gate count is about 2,061k and the design can run at
91MHz.

키워드

H.264/AVC, 인터 예측기, 참조 프레임, 가중 예측

Key word

H.264/AVC, inter predictor, reference frame, weighted prediction

* 정회원 : 한밭대학교 정보통신전문대학원

** 종신회원 : 한밭대학교 정보통신공학과 (교신저자, kkryoo@hanbat.ac.kr)

접수일자 : 2012. 07. 06

심사완료일자 : 2012. 08. 09

Open Access <http://dx.doi.org/10.6109/jkiice.2012.16.12.2725>

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits
unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서 론

H.264/AVC는 ISO/IEC 동영상 전문가 그룹과 ITU-T 비디오 코딩 전문가 그룹에 의해 표준화된 동영상 압축 기술이다. H.264/AVC는 인트라 예측 부호화, 1/4 단위의 가변적 블록 단위의 움직임 보상 및 예측, 엔트로피 부호화, 디블로킹 필터 등을 이용하여 기존의 비디오 부호화 표준인 MPEG-2 보다 약 2배, MPEG-4 보다 약 1.5배의 압축률이 향상된다[1,2]. 움직임 추정은 많은 양의 동영상 데이터 크기를 줄이기 위해 많은 동영상 압축 표준에서 폭넓게 채택하는 기술로, 연속되는 동영상 프레임들 사이의 시간적 중복성을 제거하여 압축의 효율을 높인다. 많은 양의 동영상 데이터를 다루기 때문에 계산량이 많고 외부 메모리 접근이 복잡하며, H.264/AVC 인코더의 경우 전체 연산량의 57%를 차지한다[3]. 다중 레퍼런스 프레임은 다수의 레퍼런스 프레임을 사용하여 현재 블록과 가장 유사한 블록의 위치를 움직임 벡터로 결정한다. 반복 운동이나 보호되지 않는 배경, 라이트 체인지와 같은 상황에서 레퍼런스 프레임을 하나를 사용하는 단일 레퍼런스 프레임 움직임에 비해 다수의 레퍼런스 프레임을 사용하는 다중 레퍼런스 프레임 움직임 예측 방식이 더 정확한 예측을 수행한다[4]. 계산 복잡도는 레퍼런스 프레임의 수에 비례하여 증가한다. 그러나 영상의 특성에 따라 이득은 항상 일정하지 않고, 이득 없이 많은 연산량을 소비하는 경우가 발생할 수 있다. 따라서 움직임 추정 연산에 필요한 연산량과 데이터의 양을 줄이기 위해 다양한 고속 탐색 알고리즘들이 제안되었다. 그러나 이러한 고속 탐색 알고리즘들은 외부 메모리에 대한 접근이 불규칙하고 구현 알고리즘의 계산이 하드웨어 구현에 적합하지 않은 방법들이 많다.

또한 기존의 동영상 압축 표준은 움직임 보상을 사용하여 영상의 움직임을 예측한다. 하지만 영상의 밝기를 예측하는 구조는 없었다. 그렇기 때문에 밝기가 시간적으로 변화하는 영상을 부호화 할 때 영상의 화질이 크게 열화되는 현상이 발생했다. 이러한 현상을 제거하기 위해 H.264/AVC는 레퍼런스 프레임에 적응적으로 가중치를 두어 신호를 예측하는데 사용한다[5]. 그러나 가중치 계산은 곱셈과 나눗셈을 사용하여 계산 복잡도가 높은 문제점이 있다.

본 논문에서는 하드웨어로 구현하기 적합한 목시적 가중 예측기법을 이용한 저 메모리 대역폭 인터 예측기

를 제안한다. 제안하는 예측기는 다중 레퍼런스 프레임 선택기, 목시적 가중 예측치 생성기, 참조 영역 메모리, SAD(Sum Absolute Difference) Array등으로 구성되었다.

II. 제안하는 레퍼런스 프레임 선택

본 논문에서는 하드웨어에 적합한 알고리즘을 제안한다. 제안하는 알고리즘의 순서도는 그림 1과 같다.

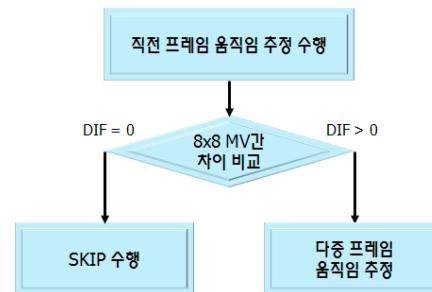


그림 1. 제안하는 레퍼런스 프레임 선택 알고리즘
Fig. 1 Proposed reference frame selection algorithm

제안하는 알고리즘은 직전 레퍼런스 프레임 움직임 추정을 수행한다. 직전 레퍼런스 프레임 움직임 추정 이후 각 블록 사이즈별 MV(Motion Vector)와 SAD값이 결정이 된다. DIF는 8x8 블록의 MV사이의 차이 값을 계산하며, 8x8 블록의 MV가 모두 동일한 경우 각각의 8x8 블록이 가려지거나 새로 나타나는 영역이 아니라 화면상에서 동일한 오브젝트일 확률이 높다. 이는 다중 레퍼런스 프레임 예측을 수행을 해도 PSNR과 비트율 성능 개선이 미미하며 직전 프레임의 움직임 추정 결과가 가장 최적의 레퍼런스 프레임일 확률이 높다. 그러므로 직전 프레임을 최적의 레퍼런스 프레임으로 결정하고, 직전 프레임의 결과로 41개의 MV들 중 오차가 가장 적은 MV를 선택하여 부호화한다. 만약 MV가 모두 동일하지 않다면, 이는 각각의 8x8 블록이 가려지거나 새로 나타나는 영역으로 화면 복잡도가 높은 경우에 해당한다. 이는 다중 레퍼런스 프레임 예측을 수행하면 PSNR과 비트율 성능이 개선될 확률이 높다.

제안된 레퍼런스 프레임 선택 구조는 그림 2와 같다. Reference Frame Selector, Current Register Buffer (CRB),

Reference Register Buffer (RRB)와 SAD Calculator Array, MV Generation, IME(Integer ME) Controller로 구성된다.

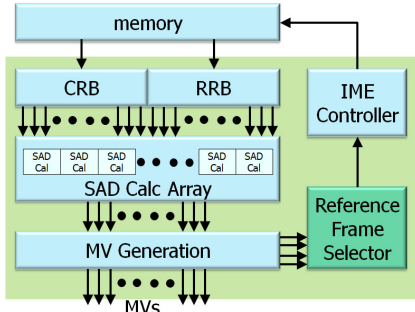


그림 2. 제안하는 레퍼런스 프레임 선택기 구조
Fig. 2 Proposed reference frame selector architecture

외부 메모리로부터 입력 받은 현재 매크로 블록과 레퍼런스 프레임 매크로 블록은 CRB, RRB에 저장된다. 저장된 현재 매크로 블록과 레퍼런스 프레임 매크로 블록은 SAD Calculator Array에서 각각 픽셀별 오차 값과 SAD값들을 계산한다. SAD는 4x4 사이즈부터 16x16 사이즈까지 블록 사이즈별로 픽셀별 오차 값의 덧셈을 통하여 값을 계산한다. MV Generation은 계산된 SAD값들과 저장된 SAD값들을 비교하여 둘 중 작은 SAD값과 MV를 저장한다. 모든 참조 영역을 계산한 이후에는 최종적으로 가장 작은 SAD값과 MV가 SAD Comparator에 저장된다.

III. 제안하는 목시적 가중 예측

가중 예측 방법은 명시적 가중 예측과 목시적 가중 예측으로 나뉜다. 명시적 가중 예측은 가중치를 부호화 중간에 결정하여 헤더에 가중치를 실어 보내는 방법으로 인코더에서 가중치를 생성해야한다. 그러나 가중치를 계산하기 위하여 모든 프레임의 데이터를 읽고 계산하는 동작을 수행해야 한다. 이는 매우 큰 연산량을 소모하며, 가중치 미적용에 비해 약 35배에 해당한다[6-7]. 이런 명시적 가중 예측은 하드웨어로 구현하기 어렵다. 목시적 가중 예측은 POC를 이용하여 프레임 사이의 거리(tb, td)를 이용하여 계산된다. 식(1)부터 식(6)까지는 JM16.0 표준에서 지정된 가중 예측 연산 수식이다. W1과 W0은

가중치를 의미한다. tb는 현재 프레임과 리스트0의 레퍼런스 프레임과의 거리이며 POC (Picture Order Count, 프레임의 출력 순서를 나타내는 값)의 뺄셈으로 값을 계산한다. td는 리스트0의 레퍼런스 프레임과 리스트1의 레퍼런스 프레임 사이의 거리이며 POC의 뺄셈으로 값을 계산한다. 시간축 상에서 현재 프레임에 가까울수록 가중치는 커지고 멀수록 가중치는 작아진다.

$$td = iClip3(-128, 128, (LIST1.POC - LIST0.POC)) \quad (1)$$

$$tb = iClip3(-128, 128, (enc.POC - LIST0.POC)) \quad (2)$$

$$tx = (16384 + iabs(td \gg 1))/td \quad (3)$$

$$DistScaleFactor = iClip(-1024, 1024, (tx * tb + 32) \gg 6) \quad (4)$$

$$W1 = DistScaleFactor \gg 2 \quad (5)$$

$$W0 = 64 - W1 \quad (6)$$

I	B	B	P	B	B	P	B	B	P	B	B	P	I
0	2	4	6	8	10	12	14	16	18	20	22	24	0

List0(1)	List1(1)	td	tb	W0	W1	List0(1)	List1(1)	td	tb	W0	W1
6	12	6	2	43	21	12	16	6	2	43	21
List0(1)	List1(3)					List0(1)	List1(3)				
6	0	-6	2	86	-22	12	6	-6	2	86	-22
List0(3)	List1(3)					List0(3)	List1(3)				
12	0	-12	-4	43	21	16	6	-12	-4	43	21

그림 3. IBBPBBP 프레임 시퀀스
Fig. 3 IBBPBBP Frame Sequence

IBBPBBP 프레임 시퀀스는 그림 3과 같다. 리스트 1과 리스트 0에 저장되어 있는 POC를 가지고 목시적 가중치를 계산한다. POC 8번과 POC 14번의 경우 리스트에 저장되어 있는 POC의 값은 다르다. 하지만 상대적인 거리는 동일하기 때문에 가중치는 동일하다. 즉 인코딩의 프레임 시퀀스가 결정되면 그에 따라 가중치가 결정되고, 가중치는 주기적으로 반복된다. 이는 가중치를 POC에 따라 복잡한 계산식을 사용하여 매번 구하는 것이 아니라 주기적인 값을 사용하게 된다. 즉 매번 계산하는 것이 아니라 ROM과 같은 저장장소에 저장해두었다가 해당 상황이 되면 해당하는 가중치 계수를 출력한다. 가중치 계수를 ROM으로 구현하여 가중치 생성을 위한 계산을 제거하여 계산 복잡도를 감소시켰으며 하드웨어 구조를 단순화 하였다. 가중치 계수를 구한 이후 식

(7)을 이용하여 참조 매크로 블록 데이터를 생성한다.

$$(Y0 * W0 + Y1 * W1 + 32) / 64 \quad (7)$$

식 (7)은 곱셈과 덧셈, 나눗셈으로 구성된다. 나눗셈은 쉬프트를 통해 간단히 구현된다. 곱셈은 Radix-4 부스 알고리즘을 사용하여 부분 곱의 합으로 계산하고, 부호 확장을 제거[8]하여 9개의 덧셈기로 구현하였다.

IV. 통합 구조 설계 및 검증

제안하는 통합 구조의 동작 흐름도는 그림 4와 같다. 시간상 가장 직전 프레임을 레퍼런스 프레임으로 단방향 IME를 수행한다. 단방향 IME는 레퍼런스 프레임을 하나만 사용하여 IME를 수행한다. 직전 프레임의 움직임 추정 이 끝난 이후 8x8 사이즈 블록의 MV를 체크해서 SKIP을 할지를 결정한다. 만약 SKIP이 되면 직전 프레임이 최적의 레퍼런스 프레임으로 선택되고 MV가 출력된다. SKIP이 발생하지 않으면 Reference RAM2에 8x8 블록 사이즈의 MV로 선택된 4개의 MV 위치의 매크로 블록 4개와 16x16 블록 사이즈의 MV로 선택된 MV 위치의 매크로 블록을 Reference RAM2에 저장한다.

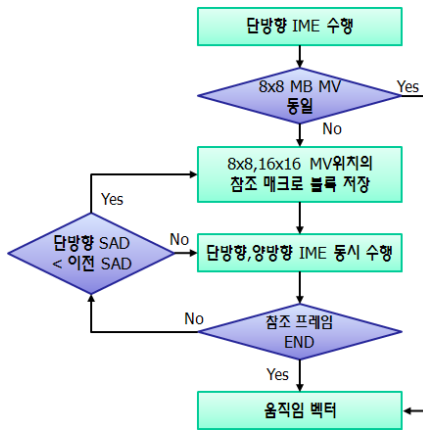


그림 4. 제안하는 통합 구조의 흐름도
Fig. 4 Proposed Integration architecture flow

Reference RAM2에 저장된 이전 프레임과 다음 레퍼런스 프레임을 레퍼런스 프레임으로 양방향 IME과 단

방향 IME을 동시에 수행한다. 양방향 IME는 목시적 가중 예측을 수행한다. 현재 레퍼런스 프레임 연산이 종료 되면 기존에 저장되어 있는 SAD값과 새로 계산된 SAD값을 비교하여 SAD가 작은 매크로 블록을 Reference RAM2에 업데이트 한다.

목시적 가중 예측은 2개의 레퍼런스 프레임을 필요로 한다. 제안하는 구조는 이전 레퍼런스 프레임 데이터 중 현재 프레임과 가장 유사한 매크로 블록만 저장하여 목시적 가중 예측을 수행한다. 이전 레퍼런스 프레임의 범위를 5개의 매크로 블록으로 제한하여 연산량을 줄이면서도 현재 프레임과 가장 유사한 매크로 블록을 사용하여 SAD가 크게 떨어지지 않는다. 또한 외부 메모리에 저장된 레퍼런스 프레임을 한번에 1개만 읽어도 목시적 가중 예측이 가능하여 외부 메모리의 대역폭을 절반으로 낮추었다. 참조 영역이 128일 경우 JM 16.0은 현재 매크로 블록 256픽셀과 참조 영역 40898픽셀을 사용하여 총 41154 픽셀을 저장할 참조 영역 메모리가 필요하다. 제안하는 구조는 현재 매크로 블록 256픽셀과 참조 영역 21729픽셀을 사용하여 총 21985 픽셀을 저장할 참조 영역 메모리가 필요하다. 제안하는 구조는 JM16.0에 비해 참조 영역 메모리가 약 46%감소하였다.

제안하는 통합 하드웨어는 IME Controller와 SAD Array, IMV Generation, Bi-prediction, Current MB RAM, Reference RAM1, 2 로 구성되며 구조는 그림 5와 같다. Reference RAM1, 2는 참조하는 매크로 블록을 저장하는 레지스터이다. Biprediction은 목시적 가중 예측기로 Reference RAM1, 2의 참조 매크로 블록과 외부 메모리로부터 읽은 참조 매크로 블록을 가지고 가중 예측을 수행하는 모듈이다. SAD array는 각각 픽셀별로 오차 값을 구하고, 픽셀별 오차 값은 블록 사이즈별 덧셈을 통하여 블록 사이즈별 SAD값을 구한다. IMV Generation은 이전에 계산했던 SAD값과 새로 계산된 SAD값을 비교하여 둘 중 작은 SAD값과 MV를 저장한다. 모든 참조 영역을 계산한 이후 최종적으로 가장 작은 SAD값과 MV가 IMV Generation에 저장되며, 저장된 MV는 FME모듈로 전달된다. Reference Frame Selector는 직전 레퍼런스 프레임의 움직임 예측이 끝나고 8x8 사이즈 블록의 MV를 체크하여 SKIP 할지를 결정한다. IME Controller는 전체 동작을 컨트롤 한다. 제안하는 구조는 병렬 구조로 6개의 SAD array로 구성되며 2가지의 모드로 동작한다. SAD Array(P only)는 항상 레퍼런스 프레임이 1개인 계

산을 수행한다. SAD Array(B)는 MUX를 통해 두 가지의 데이터중 하나를 선택하여 입력받는다. MUX 컨트롤 신호에 따라 레퍼런스 프레임이 1개인 계산을 수행하거나, 레퍼런스 프레임이 2개인 목시적 가중 예측 계산을 수행한다.

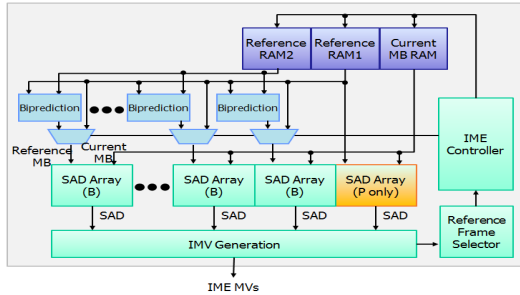


그림 5. 제안하는 통합 구조
Fig. 5 Proposed Integration architecture

제안하는 알고리즘의 성능 평가를 위해 H.264/ AVC 참조 소프트웨어 JM 16.0과 제안하는 알고리즘의 실험 결과를 비교하였다. 성능 평가를 위해 CIF 포맷 비디오 (352x288)인 Football, Crew, mobile, Paris영상, HD 포맷 비디오(1280x720)인 Stockholm, Shields영상, FullHD 포맷 비디오(1920x1080)인 Sunflower, Pedestrian area영상들이 사용되었다. 평균 0.09dB 정도의 PSNR이 감소하고 평균 180.79정도의 Bitrate가 증가하며 평균 36%정도의 Skip이 발생하였다. 이는 참조 프레임이 3이고 100프레임을 수행하여 JM 16.0은 300개의 프레임을 참조하며 제안하는 알고리즘은 228개의 프레임을 참조하므로 24%의 접근율이 감소하였다.

표 1. 알고리즘 성능 비교
Table. 1 Performance comparison of algorithm

		PSNR	Bitrate	Skip
Football (CIF)	JM 16.0	35.35	1503.36	
	Proposed	35.26	1558.98	0.22
	차이	0.09	55.62	
Crew (CIF)	JM 16.0	35.80	748.78	
	Proposed	35.62	784.83	0.39
	차이	0.18	36.05	
Mobile (CIF)	JM 16.0	32.34	2056.84	
	Proposed	32.25	2147.04	0.28
	차이	0.09	90.20	

Paris (CIF)	JM 16.0	34.67	830.13	
	Proposed	34.61	844.85	0.74
	차이	0.06	14.73	
stockholm (HD)	JM 16.0	34.17	5877.72	
	Proposed	34.14	6204.12	0.40
	차이	0.03	326.40	
shields (HD)	JM 16.0	34.62	4369.77	
	Proposed	34.47	4649.11	0.27
	차이	0.15	279.33	
Sunflower (FullHD)	JM 16.0	39.91	3581.53	
	Proposed	39.88	3738.77	0.23
	차이	0.03	157.24	
Pedestrian area (FullHD)	JM 16.0	38.96	5479.89	
	Proposed	38.80	5966.63	0.40
	차이	0.15	486.74	
Average	JM 16.0	35.72	3056	
	Proposed	35.63	3236.79	0.36
	차이	0.1	180.79	

본 논문에서 제안하는 통합 구조는 Verilog HDL로 설계 하였으며 IDEC에서 지원하는Synopsys사의 Design Compiler를 이용하여 Magnachip 0.18um CMOS 공정으로 합성된 결과 91MHz로 동작하는 것을 확인하였으며, 회로의 크기는 총 2,061,536 게이트이다.

V. 결론

본 논문에서는 H.264/AVC 인코더의 성능 향상을 위해 다중 참조 프레임 기법과 목시적 가중 예측 기법을 이용하고 낮은 외부 메모리 접근율을 위해 이전 참조 프레임 데이터를 재사용하는 인터 예측기 하드웨어 구조를 제안한다. 8x8 MV를 사용하여 Skip하는 참조 프레임 선택기와 ROM과 덧셈기로 구현한 목시적 가중치 예측기를 사용하여 하드웨어 구조를 단순화 시키고 병렬구조를 활용하여 단방향 IME와 양방향 IME를 동시에 계산한다. 제안하는 알고리즘은 JM16.0과 비교하여 평균 0.1dB 정도의 PSNR이 감소하고 전체 Bitrate 대비 증가는 약 6%정도 증가하였으나 평균 36%정도의 Skip이 발생하여 참조 프레임 접근율이 약 24%만큼 감소하였다. 참조 영역 메모리가 약 46%만큼 감소하였다. 통합 구조는 Verilog HDL로 설계되고 Magnachip 0.18um공정으로 합성된 결과 게이트 수는 약 2,061k 이고 91Mhz로 동작한다.

제안하는 하드웨어 구조는 낮은 참조 영역 메모리와 외부 메모리 접근율을 가지며 다중 참조 프레임 기법과 묵시적 가중 예측을 수행하여 고화질 영상 압축에 적합하다.

참고문헌

[1] ITU-T Recommendation H.264 and ISO/IEC 14496-10, Advanced Video Coding for Generic Audio Visual Service, May 2010.

[2] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard and Ajay Luthra, "Overview of the H.264/AVC video coding standard", IEEE Transaction on circuits and System for Video Technology, vol. 13, no. 7, pp. 560-576, July 2003.

[3] I. E. G. Richardson, H.264 and MPEG4 Video Compression, John Wiley & Sons, Dec. 2003.

[4] Y. Su and M. T. Sun, "Fast multiple reference frame motion estimation for H.264/AVC," IEEE Trans. Circuits and System for Video Technology, vol. 16, pp. 447-452, Mar. 2006.

[5] Boyce, J.M, "Weighted prediction in the H.264/MPEG AVC video coding standard," IEEE Trans. International Symposium on Circuits and Systems, vol. 3, pp. 789-792, May. 2004.

[6] Y. W. Huang, B. Y. Hsieh, S. Y. Chien, S. Y. Ma and L. G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," IEEE Trans. Circuits and System for Video Technology, vol. 16, pp. 506-522, Apr. 2006.

[7] 최지호, 선우명훈, "다양한 영상의 밝기 효과에 대해 효과적으로 적응하는 H.264/AVC의 가중치 예측 생략 방법," 전자공학회논문지, 제 47권, SP편, 제5호, pp. 965-700, Sep. 2010.

[8] 서정욱, 이상홍, "암호 프로세서용 고속 64 X 64 곱셈기," 한국통신정보보호학회 종합학술박람회 논문집, vol. 8, pp. 471-481, Dec. 1998.

저자소개



김진영(Jinyoung Kim)

2010년 2월 한밭대학교
정보통신공학과 공학사
2012년 2월 한밭대학교
정보통신공학과 공학석사

2012년 ~ 현재 (주)실리콘웍스 개발1실 연구원
※관심분야: SoC 플랫폼 설계 및 검증, 영상코덱 설계



류광기(Kwangki Ryoo)

1986년 2월 한양대학교
전자공학과 공학사
1988년 2월 한양대학교
전자공학과 공학석사

2000년 2월 한양대학교 전자공학과 공학박사
1991년~1994년 육군사관학교 교수부 전자공학과
전임강사
2000년~2002년 ETRI 시스템IC설계팀 선임연구원
2010년~2011년 Univ of Texas at Dallas 방문교수
2003년~현재 한밭대학교 정보통신공학과 교수
※관심분야: SoC 플랫폼 설계 및 검증, 하드웨어/소프트웨어 통합설계 및 검증, 멀티미디어 코덱 설계