

논문 2008-45SD-9-9

천이 수정을 통한 고 성능 테스트 데이터 압축 기법

(A New High Performance Test-data Compression Scheme Using Transition Modification)

박 재 석*, 양 명 훈*, 김 용 준*, 박 영 규*, 윤 현 준*, 강 성 호**

(Jaeseok Park, Myunghoon Yang, Yongjoon Kim, Youngkyu Park, Hyunjun Yoon, and Sungho Kang)

요 약

본 논문은 높은 압축 성능을 가진 테스트 데이터 압축 기법을 제안한다. 제안하는 기법은 돈 케어 리맵핑을 이용한 천이 추가 단계와 천이 이전 단계로 구성된다. 실험 결과는 새로운 기법이 가장 높은 성능의 기법들 중 하나인 RL-huffman보다 높은 압축률을 제공하고 더 작은 하드웨어 면적을 필요로 함을 보여준다. 따라서 이 기법은 테스트 데이터 압축을 위한 실제적인 해법으로 널리 쓰일 수 있다.

Abstract

This paper proposes a new test data compression scheme which has good performance. The proposed scheme is composed of adding transition stage and shifting transition stage using don't care remapping technique. The experimental results show that the new scheme provides higher compression ratio than RL-huffman encoding which is the one of the highest performance schemes, and requires smaller hardware overhead. Therefore it can be widely used as a practical solution for test data compression.

Keywords : RL-huffman, Test data compression, Low power test

I. 서 론

세상의 모든 제품들은 적합한 테스트를 거치게 되고 테스트에서 통과된 개체들만이 상품으로서 시장에 나올 수 있다. 반도체 산업의 핵심 제품인 칩 또한 하나의 제품이고, 테스트 과정을 거치지 않은 칩은 완제품이라 볼 수 없다. 결국 테스트 과정은 칩이라는 상품의 시장 적기 출하와 생산 비용에 있어서 매우 중요한 요소 중 하나이다. 테스트 과정을 위해서 테스트 패턴이 필

요한데, 이 패턴은 ATE (Automatic Test Equipment) 으로부터 보내어진다. 이때 테스트 데이터를 전송하는 데 걸리는 시간이 시장 적기 출하 면에서 가장 큰 이슈이다. 오늘날, 칩의 복잡도가 급상승하고 있기 때문에 우리는 좀 더 거대한 양의 테스트 데이터가 수율을 높이기 위해 필요하다. 여기서 데이터를 보내는 속도에는 한계가 있으므로, 데이터의 양을 줄이는 압축기법의 중요성이 높아지고 있다.

이를 위해 다수의 데이터 압축 기법들이 제안되어 왔다. 이 기법들은 특성에 따라 여러 갈래로 나뉘어져 분석되는데, 대표적인 갈래로 런랭스 기반의 기법들이 있다. 콜롬 코딩^[1]은 각 구간의 연속된 0의 길이를 그대로 인코딩하여 압축을 수행하고, FDR (frequency-directed run length) 코딩^[2]과 확장된 FDR 코딩^[3]은 짧은 길이에 짧은 코드워드를 인코딩하여 좀더 좋은 압축률을

* 학생회원, ** 평생회원, 연세대학교 전기전자공학과 (Department of Electrical Electronic Engineering, Yonsei University)

※ 이 논문은 2004(2005)년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임 (No. R01-2006-000-11038-0).

접수일자: 2008년3월26일, 수정완료일: 2008년8월26일

..... 0 11X 0X 1X1X1 0XX 1XX 0XX 1XX1X 0XX 1X 0XX0X 1
 0 111 00 11111 000 1111 000 11111 000 11 00000 1
 3 2 5 3 4 3 6 3 2 5

그림 1. RL-허프만 인코딩 예시
 Fig. 1. An example of RL-huffman encoding.

보인다. 보통 런랭스 기반 압축 방식은 작은 디코더 면적을 필요로 하지만, 비교적 압축 성능이 떨어진다. 또 다른 대표적 갈래로서 통계적 방법들이 있다. SC (statistical coding)^[4]는 정해진 길이의 구간들을 통계적으로 분석, 빈도수에 따라 코드워드를 배정하여 인코딩을 수행하고, 선택적 허프만 코딩^{[5], [6]}은 빈도수가 낮은 구간에 대해서는 코드워드를 부여하지 않아 하드웨어 면적면에서 좋은 효과를 보여준다. 또한 VIHC (variable-input huffman coding)^[7]에서는 구간 길이가 변할 수 있어서 높은 압축률을 보여주고, 나인 코드드 기법^[8]에서는 돈 케어에 대하여 인코딩을 수행, 고정된 코드워드를 가지고 있다. 통계적 방법들은 좋은 압축률을 보이지만, 비교적 큰 디코더 면적을 필요로 한다. 최근 압축 방식에는 칩의 복잡도가 급상승하면서 하드웨어 면적보다는 압축 성능에 좀 더 무게를 두고 있다. 이에 위에서 언급된 두 갈래의 특성을 융합하여 스캔셀에서의 전력 소비를 최소화 시키고, 압축 성능을 극대화한 기법이 RL-허프만 (mixed run-length and huffman) 코딩^[9]이다.

그림 1은 RL-허프만의 예시를 보여준다. 이 방식은 기존의 다른 방식들과 비교하여 테스트 스캔 셀에서의 소비전력을 최소화하기 위하여 돈 케어 비트 맵핑에서 천이가 최대한 적게 그리고 뒤에서 일어나도록 설정한다. 맵핑 후, 런랭스 개념을 이용하여 구간을 나눈 후, 각 구간의 길이를 통계적으로 분석하여 코드워드를 배분하게 된다. 코드워드 배분에서 허프만 개념이 사용되어 압축률이 아주 뛰어나다. 다만 이를 위한 디코더 면적이 다소 크게 필요하다는 것이 단점이다. RL-허프만 방식에서는 디코더를 코드워드에 대한 룩업 테이블을 이용하여 구성하므로, 결국 코드워드의 개수가 디코더의 크기를 결정한다.

이 점으로부터 본 논문에서는 좀 더 효과적인 압축 성능과 축소된 하드웨어 면적을 가진 기법을 제안한다. 본 논문은 다음과 같이 구성된다. 우선 II장 본론에서는 제안하는 기법의 과정에 대하여 설명을 하며, III장 실험 부분에서는 각 방법에 대한 실험 결과와 다른 압축 방식과의 비교 실험을 수행하도록 하겠다. 마지막으로 IV장에서는 결론을 맺는다.

II. 본 론

다음에서 제안하는 기법은 생성된 테스트 패턴 중에서 돈 케어 비트는 0이나 1 어느 값으로든 맵핑이 가능하다는 성질을 이용하여 압축률 성능을 올리고, 하드웨어 크기를 줄인다.

그림 2의 순서도는 제안된 압축 기법의 전체적인 과정을 나타낸다. 먼저 RL-huffman 기법과 같은 방법으로 돈 케어 맵핑을 하고, 각 구간에 대한 통계적인 분석을 수행한다. 이 분석에서 제안된 기법에서 필요한 자료를 얻고, 천이 수정 기법을 수행한다. 천이 수정 기법에서 각 구간을 좀 더 효과적으로 돈 케어 리맵핑을 통해 구성한 후, 다시 각 구간에 새로운 코드워드를 부여하고 인코딩과정을 거쳐 압축이 이루어진다. 천이 수정 과정을 좀 더 자세히 살펴보면 테스트 데이터의 첫 번

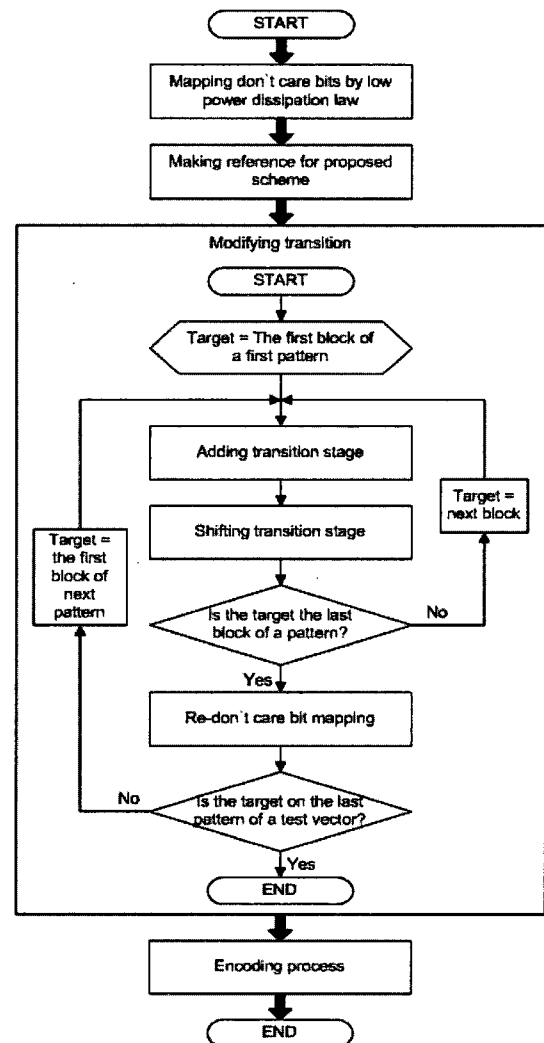


그림 2. 제안된 압축 과정
 Fig. 2. The proposed compression process.

제 테스트 패턴부터 각 구간을 살펴가면서 두 단계의 기법을 거치고 난 후 다시 돈 케어 리맵핑을 수행한다. 여기서 거치게 되는 두 단계는 천이 추가 단계와 천이 전이 단계로 밑에서 자세히 살펴보도록 한다.

1. 천이 추가

이 과정은 코드워드 개수를 줄이고 곧 하드웨어 면적을 줄이기 위한 긴 비트 길이 값을 가진 구간을 3개의 구간으로 나누는 과정이다. 즉, 돈 케어 맵핑을 이용하여 비트천이 두 개를 추가하는 것이다. 이로 인해 하드웨어 크기를 감소시킬 수 있다.

그림 3는 천이 추가 단계의 과정이다. 우선 대상이 천이 추가가 가능한지를 살펴보고, 가능하다면 대상을 3 구간으로 나누어 가장 좋은 경우를 찾아내는 흐름을 보여준다.

그림 4는 비트 천이 애딩에 대한 하나의 예시이다.

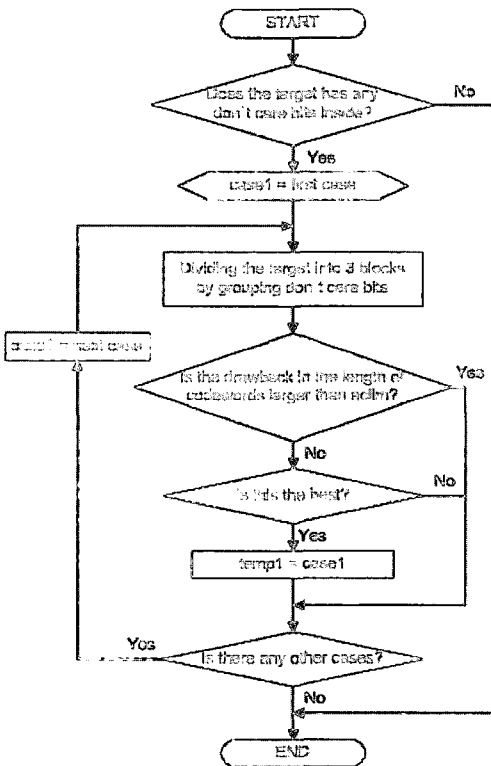


그림 3. 천이 추가 단계
Fig. 3. Adding transition stage.

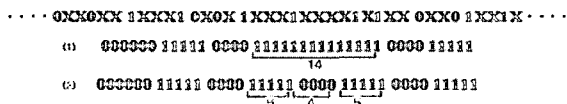


그림 4. 천이 추가 예시
Fig. 4. An example of adding transition position

(1)은 RL-허프만의 돈 케어 맵핑 방식을 보여주고 있다. 이제 패턴의 각 구간을 살펴보면서 (2)와 같이 3개의 새로운 구간으로 나눌 수 있는 구간을 찾게 되는데, 이 제안된 방법을 쓸 수 있는 조건은 다음과 같다.

조건 1-1 : 대상 구간 양 끝 비트들을 제외하고 다른 비트 들 중에 돈 케어 비트가 있어야 한다.

조건 1-2 : 3개로 나눈 구간들에 배정되는 코드워드 길이의 합이 나누어지기 전의 코드워드 길이 보다 일정 비트길이 이상 길어지는 것은 불가하다.

예시에서는 4, 5, 6 길이의 구간은 높은 빈도수를 가지고 짧은 코드워드 길이를 가진다고 가정하겠다. 따라서 대상 구간인 14 길이의 구간에서 먼저 양 끝 비트를 제외한 다른 비트 중에 돈 케어 비트가 있으므로 조건 1-1을 만족하고, 돈 케어 비트 구간에 0 구간을 삽입함으로써 구간을 3개로 나눌 수 있음을 알 수 있다. 그리고 조건 1-2를 만족시키는지 확인 한다. 이런 조건을 통과한 경우 중에서 가장 코드워드 길이가 짧고 천이가 뒤에서 일어나는 경우를 채택하게 된다. 천이가 뒤에서 일어난다는 것은 전력 손실이 그 만큼 적다는 것을 의미하는데 실험 부분에서 논의 하도록 하겠다.

시뮬레이션에서는 코드워드 길이의 증가분을 그림 3의 하나의 과정에서 볼 수 있듯이 aclim (Additional Codeword Limit : 추가 허용 코드워드 길이) 이라는 변수로 제한을 두고 있는데, 압축률에서 손해를 적게 보기 위해서는 그 값을 적게 해주어야 하고, 코드워드의 개수를 줄이는 데에 주목하기 위해서는 큰 값을 주어야 한다. 이 또한 실험에서 살펴보도록 하겠다.

2. 천이 이전

위에서 언급 했듯이 전력 손실을 최소로 한 돈 케어 맵핑은 천이를 가장 뒤에서 일어나도록 한다. 천이 위치를 좀 더 효율적으로 바꿔주게 되면 압축률과 하드웨어 크기 면에서 좋은 결과를 얻을 수 있다.

그림 5는 천이 이전 단계의 과정을 보여준다. 먼저 대상이 천이 이전이 필요한 구간인지를 살펴보고, 필요하다면 천이 이전이 가능한지 살펴본 후, 돈 케어 맵핑을 바꿈으로써 천이를 왼쪽으로 이동시킨다.

그림 6은 비트천이 이전에 대한 예를 보여주고 있다. (1)은 역시 RL-허프만의 돈 케어 맵핑 방식을 보여주고 있다. 이전 천이 추가 예시와 같이 빈도수 체크 결과가 비트 길이 3, 4, 5, 6을 가지고 있는 구간의 빈도수는

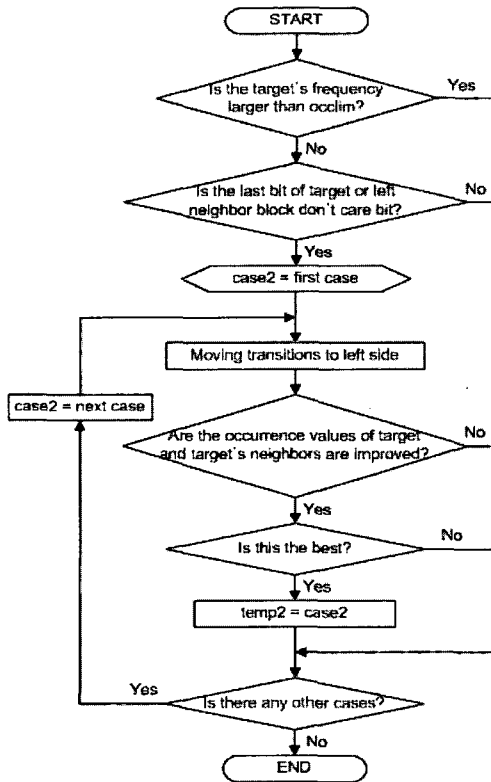


그림 5. 천이 이전 단계
 Fig. 5. Shifting transition stage.

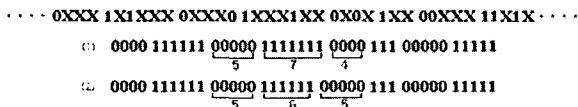


그림 6. 비트 천이 쉬프팅 예시
 Fig. 6. An example of shifting transition position.

높고, 비트 길이 7의 구간 빈도수는 낮다고 가정을 했을 때, 비트 7 구간은 (2)와 같이 돈 케어 맵핑에 수정을 가하여 천이구간을 옮김으로써 비트 길이 6을 가지게 할 수 있다. 이 기법 사용을 위해서는 다음의 조건들을 만족시켜야 한다.

조건 2-1 : 대상 구간 바로 이전 비트들과 대상 구간 마지막 비트들 중에 적어도 한 쪽은 돈 케어 비트이어야 한다.

조건 2-2 : 대상 구간은 적어도 같거나 더 큰 빈도수를 가지는 길이의 구간으로 바뀌어야 한다.

조건 2-3 : 대상 구간의 길이를 변화시키면서 생기는 인접 구간의 길이 변화를 확인, 높은 빈도수를 가졌던 구간이 낮은 빈도수를 가지게 되는 것은 불가하다.

이전 예시에서 대상 구간 이전 비트는 돈 케어 비트가 아니지만, 대상 구간 가장 마지막 비트가 돈 케어 비

트이기 때문에 조건 2-1을 만족시킨다. 또한 대상 구간의 길이가 7에서 6으로 더 높은 빈도수의 구간으로 변하였기 때문에 조건 2-2를 만족시키며, 길이 7이었던 구간이 6의 길이를 가지게 되면서 인접 오른쪽 구간이 4에서 5의 길이를 가지게 되는 데, 4 또한 높은 빈도수로 고려되므로 여기서 마지막 조건 2-3을 만족시키게 된다. 이런 방식으로 각 구간의 길이가 낮은 빈도수를 가지는 길이일 때 마다, 먼저 돈 케어비트의 유무(조건 2-1)을 따지고 만족한다면 변화된 대상 구간의 빈도수를 체크해서 조건 2-2를 만족하게 되면 다음으로 인접 구간들의 빈도수 변화를 보게 된다. 여기서 마지막 조건 2-3을 만족시키면 돈 케어 리맵핑이 일어나게 된다. 물론 일어날 수 있는 모든 경우를 체크해보고 가장 빈도수를 높일 수 있는 방법을 찾게 된다.

제안된 방식에서는 낮은 빈도수와 높은 빈도수를 나누는 기준이 되는 특정한 값이 필요하게 되는데, 그림 5의 조건에서의 값이 시뮬레이션에서는 어느 정도의 빈도수까지를 낮은 빈도수로 고려할 것인지를 occlim (Low Occurrence Limit : 낮은 빈도 기준 값)이라는 변수를 가지고 설정하게 된다.

3. 한 기법으로서의 두 과정 과 비용

먼저 첫 번째 제안된 천이 추가 과정으로 긴 구간을 짧은 구간으로 나누어주면, 압축률을 다소 손해 보면서 코드워드 개수를 줄이게 된다. 이후 두 번째 천이 이전 과정을 통해 천이위치를 옮겨서 압축률을 다시 보상하고 코드워드 개수는 추가적으로 줄이게 된다. 첫 과정으로 인한 압축률 손실은 두 번째 과정에서 얻는 성능이득에 비하여 아주 작다. 따라서 제안된 기법은 압축률과 하드웨어 면적 두 가지 면에서 큰 효과를 볼 수가 있다. 물론, 스캔 셀의 전력 손실은 더 커지지만 실험결과는 다른 압축방법에서의 맵핑 방식보다는 여전히 작은 전력을 소비함을 보여준다. 두 과정을 위해서는 컨트롤 값으로 위에서 언급한 두 가지 컨트롤 값(occlim, aclim)이 모두 필요하게 된다. 이 값에 따른 결과의 차이 또한 실험 부분에서 논할 것이다. 디코더 구조는 RL-허프만 디코더 구조와 차이가 없다. 단지 제안된 기법에서는 코드워드 개수가 줄어들어 그 만큼의 하드웨어 면적을 줄일 수 있다.

III. 실험결과 및 분석

실험을 통하여 위에서 제안한 압축 기법과 RL-허프

만 그리고 기존의 압축방식들의 성능을 비교한다. 여기서 사용되는 테스트 패턴은 테트라맥스^[10] 프로그램을 이용하여 생성하였다.

표 1은 첫 번째 제안한 과정, 천이 추가 단계에 대한 결과이다. 각 추가 허용 코드워드 비트 수(aclim)가 증가함에 따라서 세 개로 갈라질 수 있는 구간의 개수가 증가하게 되는데, 그에 따라서 코드워드의 개수가 줄어들고, 압축률은 조금씩 더 나빠지고 있음을 알 수 있다. 컨트롤 값 aclim을 크게 설정하면 그 손실은 더 커지는데, 반대로 하드웨어 면적을 나타내는 코드워드 개수는 계속 줄어든다. 하지만 여기서의 압축률 손실은 두 번째 과정에서 보상받게 된다. 표에서 마지막 열에 있는 값은 제안된 단계를 통합함으로써 추가되는 가중천이

표 1. 코드워드 추가 허용 비트 수에 따른 천이 추가
Table 1. Adding transition in case of several additional codeword bits.

Circuit	aclim = 0			aclim = 1		
	CR	#CW	+%WT	CR	#CW	+%WT
s5378	56.28	64	0.05	56.14	61	4.98
s9234	47.25	65	0.00	47.25	63	0.10
s13207	69.51	165	0.00	69.51	165	0.00
s15850	57.83	110	0.00	57.83	109	0.39
s35932	55.28	124	8.91	55.07	117	12.89
s38417	89.44	175	0.00	89.32	168	13.94
s38584	61.52	212	0.00	61.52	209	0.06
Circuit	aclim = 2			aclim = 3		
	CR	#CW	+%WT	CR	#CW	+%WT
s5378	55.41	50	25.41	55.32	40	43.29
s9234	45.27	56	16.26	43.55	48	56.46
s13207	69.51	159	0.36	69.47	149	1.10
s15850	57.79	94	1.57	55.18	73	74.02
s35932	54.58	109	16.70	54.08	104	18.74
s38417	89.11	158	27.40	88.52	135	64.75
s38584	61.49	199	1.01	58.92	168	64.62

표 2. 압축률 최적화 천이 이전
Table 2. Shifting transition position for high compression ratio.

Circuit	[9] RLhuff. CR(%)	occlim	Prop. CR(%)	# of Codewords		+%WT
				Origin.	Prop.	
s5378	56.27	13	57.66	65	56	0.40
s9234	47.25	703	49.10	65	61	0.29
s13207	69.51	847	70.92	165	119	0.23
s15850	57.83	1061	59.11	111	90	0.15
s35932	55.08	3394	55.51	127	108	0.01
s38417	89.44	3517	89.63	177	165	0.03
s38584	61.52	3633	63.25	212	156	0.12

(Weighted Transition)^[11] 값을 나타낸 것이다. 이는 곧, 스캔셀에서 소모될 추가 전력을 나타낸다. 모든 벤치 회로에서 aclim값을 크게 설정했을 때는 그 증가 비율이 무시 못 할 값을 가지는데, 실제 제안하는 기법에서는 작은 aclim값만 사용한다. 뒤에서 실제 사용되는 값들은 다른 맵핑에 비하여 무시할 정도의 값이라는 것을 보이도록 하겠다.

표 2와 표 3은 첫 번째 단계를 생략한 두 번째 과정인 비트 천이 이전 단계만의 결과이다. 표 1은 압축률이 높게 나오는 결과를 모은 것이고, 표 2는 코드워드 개수를 최소로 만드는 결과를 모은 것이다. 두 경우 모두 압축률의 향상과 코드워드 개수가 줄어드는 현상을 관찰 할 수 있다. 압축률을 최적화 시키는 낮은 빈도수 기준 값(occlim)은 많은 회로에서 아주 큰 값을 가짐을 알 수가 있는데, 실험 분석 중에 이 값들은 구간들의 빈도수 중 가장 높은 빈도수 값으로 관찰되었다. 반면에 하드웨어 면적을 최적화 시키는 occlim값은 대부분 낮은 경향을 가진다. 역시 압축률 최적화시보다는 더 낮은 압축률을 가지고 있지만 여전히 RL-허프만보다는 더 높은 압축률을 가지고 있다. 천이 이전 단계도 천이를 왼쪽으로 옮김으로써 추가 전력 소비가 나타나는데, 천이 추가 단계 보다는 아주 작은 추가적인 전력 소비

표 3. 하드웨어 면적 최적화 천이 이전
Table 3. Shifting transition position for small hardware overhead.

Circuit	[9] RLhuff. CR(%)	occlim	Prop. CR(%)	# of Codewords		+%WT
				Origin.	Prop.	
s5378	56.27	5	57.06	65	53	0.24
s9234	47.25	8	47.75	65	54	0.11
s13207	69.51	847	70.92	165	119	0.23
s15850	57.83	70	58.69	111	86	0.12
s35932	55.08	15	55.35	127	107	0.01
s38417	89.44	2	89.45	177	150	0.01
s38584	61.52	39	62.55	212	151	0.09

표 4. 압축률 최적화를 위한 제안된 기법
Table 4. Proposed scheme for high compression ratio.

Circuit	[9] RLhuff. CR(%)	occlim/a clim	Prop. CR(%)	# of Codewords		+%WT
				Origin.	Prop.	
s5378	56.27	14/0	57.56	65	56	0.43
s9234	47.25	703/0	49.10	65	61	0.29
s13207	69.51	847/0	70.88	165	122	0.23
s15850	57.83	968/1	59.10	111	87	0.53
s35932	55.08	3394/0	55.46	127	115	8.91
s38417	89.44	3517/0	89.62	177	164	0.03
s38584	61.52	3633/0	63.11	212	156	0.10

표 5. 하드웨어 면적 최적화를 위한 제안된 기법
Table 5. Proposed scheme for small hardware overhead.

Circuit	[9]	occlim/a clim	Prop. CR(%)	# of Codewords		+%WT
	RLhuff. CR(%)			Origin.	Prop.	
s5378	56.27	6/1	56.82	65	53	5.23
s9234	47.25	7/0	47.75	65	54	0.11
s13207	69.51	2/3	69.73	165	120	1.17
s15850	57.83	9/2	58.22	111	77	1.64
s35932	55.08	1/1	55.14	127	104	12.89
s38417	89.44	1/0	89.45	177	149	0.01
s38584	61.52	121/2	62.61	212	145	1.10

표 6. 돈 케어 맵핑에 따른 weighted transition 비교
Table 6. Comparison of weighted transition for different don't-care-mapping skills.

Circuit	[9]	Zero	Random
	RL-huffman		
s5378	100	139	479
s9234	100	175	405
s13207	100	154	776
s15850	100	154	503
s35932	100	120	292
s38417	100	165	2133
s38584	100	145	569

가 결과에서 나타난다. 천이 추가는 새로운 천이들의 가중 천이 값이 필요하지만, 천이 이전은 옮긴 위치만큼의 추가 전력 소비만 필요하기 때문이다.

표 4와 표 5는 두 과정을 차례대로 거친 제안된 압축 방안의 결과를 보여준다. 이전의 첫 번째 기법의 결과와 마찬가지로 압축률과 코드워드 개수 면에 있어서 최적의 결과들을 따로 모아보았다. 단, 코드워드 최적의 결과는 압축 성능이 RL-허프만보다 나빠지지 않는 선에서 추출하였다. 모든 실험 대상 회로에서 RL-허프만보다 높은 압축률과 적은 코드워드 개수를 확인할 수 있다.

이상의 기법들에서 다른 성능을 위하여 추가적으로 필요한 비용요소는 스캔셀에서의 전력 손실이다. 각 결과에서 그 값을 RL-허프만에 비교하여 퍼센트 값으로 나타내었다. 표 6은 RL-허프만의 가중천이 값을 100으로 놓았을 때, 다른 돈 케어 맵핑에서의 그 상대 값을 보여주고 있다. 우선 랜덤 맵핑은 전 구역에서 아주 큰 상대 값을 보여주고 있다. 그리고 0으로 맵핑한 기법도 평균적으로 약 40%정도의 추가 가중 천이 값을 가지고 있음을 볼 수 있겠다. 따라서 위 제안한 기법의 결과 값에서 필요로 하는 총 전력 값은 다른 맵핑 방법을 쓰든 타 압축 기법에 비하여 여전히 훨씬 작은 값을 갖는다

표 7. 제안된 방식과 다른 압축 방식들과의 비교
Table 7. Comparison among proposed scheme and other compression techniques.

Circuit	[2] FDR	VIHC[7]				Prop.	
		Group Size		Max	#CW	CR	#CW
		8	16				
s5378	50.50	44.77	49.13	54.93	64	57.56	56
s9234	40.10	41.18	41.86	43.15	63	49.10	61
s13207	69.59	66.26	67.48	71.15	183	70.88	122
s15850	56.82	54.28	54.99	58.99	110	59.10	87
s35932	44.07	35.10	41.99	51.08	118	55.46	115
s38417	85.17	75.95	80.08	86.90	404	89.62	164
s38584	60.84	58.43	59.28	62.33	212	63.11	156

는 것을 알 수 있다. 뒤에서 제안된 기법과 비교될 FDR과 VIHC는 돈 케어 맵핑을 압축률을 높게 얻을 수 있는 0 맵핑 방식을 사용하였다. 따라서 이 무시할 정도의 비용으로 압축률과 하드웨어 면에서 이득을 볼 수 있으므로 이 기법은 효율이 좋다고 할 수 있다.

마지막으로 표 7은 제안된 방식 결과 중 표4의 결과를 이전 대표적인 압축 방식과 비교한 결과이다. 런타임 기반 압축 방식인 FDR보다 제안된 방식이 확연히 좋은 압축 성능을 가졌음을 우선 한눈에 볼 수 있다. 이제 같은 성격을 가지고 있는 VIHC를 살펴보기 위하여, 먼저 코드워드의 개수를 적게 가질 수 있는 그룹 크기 설정 방법을 사용, 그룹 크기를 16까지 설정했을 경우에는 제안된 방식과 압축률 차이가 꽤 많이 나고 있다. 압축률을 가장 높일 수 있는 그룹 크기를 정해주지 않았을 경우에는 단 s13207의 경우에만 VIHC가 근소한 차이로 더 좋은 압축률을 보인다. VIHC 기법이 RL-허프만의 결과와는 더 큰 차이를 보이는 것을 이전 표들에서 확인할 수 있는데, 제안된 기법이 그 차이를 크게 줄였다. 그리고 결정적으로 표에서 그룹크기를 설정하지 않았을 때 VIHC에서 필요로 하는 코드워드 개수를 볼 수가 있는데, 모든 회로에서 제안된 방법보다 훨씬 더 많은 코드워드가 필요하다. 특히 VIHC가 압축 성능이 좋은 s13207의 경우에는 코드워드 개수에서 큰 차이가 나고 있다. 나머지 회로에서는 제안된 방법이 VIHC보다 더 좋은 압축률을 보여준다.

실험 결과를 종합해서 보면 첫 번째 과정을 거치게 되면서 코드워드는 더 줄어들고 몇몇 회로에서 아주 미세한 압축률 손실이 일어나는데, 만약 하드웨어는 고려하지 않고 압축률만을 중요시 한다면 첫 번째 단계를 생략하고 두 번째 단계만을 사용하여 제안된 기법을 수행할 수 있다.

IV. 결 론

본 논문에서는 RL-허프만 압축 기법이 높은 압축 성능을 가졌지만 비교적 큰 하드웨어 오버헤드를 가지고 있는 점에 주목하여 더 높은 성능과 좀 더 작은 하드웨어 면적을 가지는 기법을 제안하였다. 제안된 기법에서는 천이 추가 단계를 거쳐 하드웨어 면에서 향상을 얻은 후, 천이 이전 기법을 통하여 압축률과 하드웨어 면에서 큰 향상을 보여준다. 그 비용인 추가 전력 소모는 다른 돈케어 맵핑 방식에 비해서 무시할 수 있는 값이나 혹은 작은 값이다. 반면, 제안된 방식의 압축 성능은 타 압축 방식에 비해서 크게 높은 결과를 보여준다. 따라서 제안된 방식은 효과적인 고 성능 압축 기법이라 할 수 있다. 좀 더 고집적 반도체 산업으로 나아가는 현대의 추세에서 압축률의 중요성이 더 높아지고 있으므로 위 방안은 유용하게 쓰일 수 있을 것으로 기대된다.

참 고 문 헌

- [1] A. Chandra, and K. Chakrabarty, "System-on-a-chip test data compression and decompression architectures based on Golomb codes," *IEEE Trans. Computer-Aided Design*, vol. 20, pp. 113 - 120, Mar. 2001.
- [2] A. Chandra, and K. Chakrabarty, "Frequency-Directed Run-Length (FDR) codes with application to system-on-a-chip test data compression," in Proc. IEEE VLSI Test Symp., Apr. 2001, pp. 114-121.
- [3] A. El. Maleh and R. Al-Abaji, "Extended frequency-directed run-length codes with improved application to system-on-a-chip test data compression," in Proc. Int. Conf. Electronic Circuits Systems (ICECS'02), 2002, pp. 449 - 452.
- [4] A. Jas, J. Ghosh-Dastidar, and N. A. Touba, "Scan vector compression/decompression using statistical coding," in Proc. IEEE VLSI Test Symp., Apr. 1999, pp. 114-121.
- [5] A. Jas, J. Gosh-Dastidar, M. Ng, and N. Touba, "An efficient test vector compression scheme using selective Huffman coding," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 22, no. 6, pp.797 - 806, Jun. 2003.
- [6] X. Kavousianos, E. Kalligeros., and D. Nikolos, "Optimal selective huffman coding for test-data compression," *IEEE Transactions on Computers*, Volume 56, Issue 8, pp. 1146-1152 Aug. 2007
- [7] P. Gonciari, B. Al-Hashimi, and N. Nicolici, "Improving compression ratio, area overhead, and test application time for system-on-a-chip test data compression/decompression," design, Automation and Test in Europe (DATE'02), pp. 604-611, Mar. 2002.
- [8] M. Tehranipour, M. Nourani, and K. Chakrabarty, "Nine-Coded Compression Technique for Testing Embedded Cores in SoCs," *IEEE Trans. VLSI Systems*, vol. 13, pp. 719-731, June 2005.
- [9] M. Nourani, and M. Tehranipour, "RL-Huffman encoding for test compression and power reduction in scan application," *ACM Trans. Design Automat. Electron. Syst.*, vol. 10, no. 1, pp. 91 - 115, Jan. 2005.
- [10] TetraMax Reference Manual. Release 2004. 12, Synopsys Inc., Mountain View, CA, 2001.
- [11] A. Chandra, and K. Chakrabarty, "Low-Power Scan Testing and Test Data Compression for System-on-a-Chip," *IEEE Trans. On Computer-Aided Design (TCAD)*, vol. 21, no. 5, pp. 597-604, May. 2002.

저 자 소 개



박 재 석(학생회원)
 2008년 2월 연세대학교
 전자공학과 학사 졸업.
 2008년 현재 연세대학교 전기전자
 공학과 석박통합과정.
 <주관심분야 : Compression, SoC
 테스트>



양 명 훈(학생회원)
 1996년 2월 연세대학교 공과대학
 전기공학과 학사 졸업.
 1998년 연세대학교 공과대학
 전기전자공학과 석사졸업.
 2004년 삼성전자 시스템 LSI 선임
 연구원.
 2008년 현재 연세대학교 공과대학
 전기전자공학과 박사과정.
 <주관심분야 : SoC 설계, SoC 테스트>



김 용 준(학생회원)
 2002년 2월 연세대학교 공과대학
 전기공학과 학사졸업.
 2004년 2월 연세대학교 공과대학
 전기전자공학과 석사졸업.
 현 재 연세대학교 공과대학
 전기전자공학과 박사과정.
 <주관심분야 : SoC 설계, SoC 테스트>



박 영 규(학생회원)
 2004년 2월 호서대학교 공과대학
 전자공학과 학사졸업.
 2007년 2월 연세대학교 공과대학
 전기전자공학과 석사졸업.
 2008년 현재 연세대학교 공과대학
 전기전자공학과 박사과정.
 <주관심분야 : SoC 설계, SoC 테스트>



윤 현 준(학생회원)
 2004년 2월 연세대학교 공과대학
 전기전자공학과 학사졸업.
 2008년 현재 연세대학교 공과대학
 전기전자공학과 석사과정.
 <주관심분야 : SoC 설계, SoC 테
 스트>



강 성 호(정회원)
 1986년 2월 서울대학교 공대
 제어계측공학과 학사졸업.
 1988년 5월 The University of
 Texas at Austin 전기 및
 컴퓨터 공학과 석사졸업.
 1992년 5월 The University of
 Texas at Austin 전기 및
 컴퓨터공학과 박사 졸업.
 1992년 미국 Schlumberger, Research Scientist.
 1994년 미국 Motorola, Senior Staff Engineer.
 2008년 현재 연세대학교 전기전자공학과 교수.
 <주관심분야 : SoC 설계, SoC 테스트>