

HDL 코딩 방법에 따른 FPGA에서의 성능 실험 및 평가

이상욱, 이보선, 이승은*, 서태원
고려대학교 컴퓨터교육과
*서울과학기술대학교 전자정보공학과

e-mail: lesa@korea.ac.kr, l2bs@korea.ac.kr, seung.lee@seoultech.ac.kr, suhtw@korea.ac.kr

FPGA Performance Evaluation According to HDL Coding Style

Sangwook Lee, Boseon Lee, Seungeun Lee*, and Taeweon Suh
Dept. of Computer Science Education, Korea University

*Dept. of Electronic & Information Engineering, Seoul National Univ. of Science and Technology

요 약

FPGA는 대용량의 게이트를 지원하는 하드웨어를 프로그램 할 수 있는 디바이스이다. ASIC을 위해 설계된 로직은 칩으로 제조되기 전에 검증 과정을 거친다. 이 검증 과정에서 시뮬레이션의 한계를 극복하기 위해 FPGA를 사용한 에뮬레이션 방법을 많이 채택한다. 에뮬레이션 과정에서 ASIC의 동작 속도로 검증하는 것이 바람직하지만 FPGA의 특성상 ASIC과 같은 속도로 동작하기는 쉽지 않은 것이 현실이다. 본 논문에서는 HDL 코딩 방법에 따른 FPGA의 성능 민감도를 실험하였다. 실험 및 평가를 위해 다양한 알고리즘을 가진 가산기를 이용하였고 각 가산기 종류와 비트수에 따라 Verilog-HDL을 이용하여 코딩하였으며 대표적인 FPGA 제조사(Altera와 Xilinx)별, 디바이스별로 동작 속도와 자원 사용량을 측정하였다. 실험 결과 FPGA 제조사별로 다른 경향을 보임을 확인하였다. 성능 면에서는 비트별로 다소 차이는 있지만 Altera 디바이스에서는 Ripple Carry, Carry Lookahead 가산기보다 Prefix 가산기의 성능이 우수하게 나왔다. Xilinx 디바이스에서는 예상과 달리 가산기들 사이의 성능 차이가 크게 나지 않았으며 Ripple Carry, Carry Lookahead 가산기가 Prefix 가산기보다 높은 성능을 보이는 경우도 있었다. 비용 면에서는 디바이스별로 큰 차이가 나지 않았으며 ASIC과 비슷한 성능 민감도를 보였다. 그리고 각 제조사에서 제공하는 IP(Intellectual Property) Core를 사용했을 경우는 대부분의 디바이스에서 우수한 성능을 보여 주었다. TSMC 90nm 공정 기술로 제작한 ASIC과 IP Core를 비교했을 때는 ASIC의 성능이 4배 정도 우수한 것으로 나타났다.

1. 서론

반도체 분야에서는 사용자가 특정 용도의 반도체를 주문하면 제조사가 이에 맞춰 설계 및 제작을 하는 주문형 반도체, 즉 ASIC(Application Specific IC)이 많이 사용되고 있다. 제조사는 설계된 하드웨어를 ASIC으로 만들기 전에 최종적으로 동작 및 성능을 검증하는 과정이 필요한데 일반적으로 시뮬레이션 과정과 FPGA를 이용한 에뮬레이션 과정을 거친다. 두 검증 방법 중 시뮬레이션은 실제 하드웨어에 비해 너무 느린 속도로 동작을 하기 때문에 일부 기능의 단순한 테스트에만 사용하고 있으며 동작 및 성능의 종합적인 검증을 위해서는 FPGA에 의한 에뮬레이션을 많이 사용하고 있다. 에뮬레이션을 할 때에는 검증의 효율성과 정확성을 높이기 위해 FPGA가 ASIC의 속도와 비슷하게 동작하도록 하는 것이 중요한데 사용자가 내부 회로를 재구성 할 수 있도록 되어 있는 FPGA의 구조적인 특성으로 인해 FPGA가 ASIC의 속도로 동작한다는 것은 현실적으로 어려움이 따른다. 예를 들면, Schelle[3]는 인텔의 Nehalem 프로세서를 Xilinx의 Virtex 4와 Virtex 5로 구성된 다중 FPGA 에뮬레이션 시스템에 구현을 하였는데 FPGA로 합성된 Nehalem은 약 500KHz의 클럭 주파수로 동작을 하면서 x86에서 할 수 있는 일반적인 작업들을 수행하였다. 그러나 대부분의 Nehalem

프로세서들이 2GHz 이상의 클럭 주파수로 동작한다는 것을 감안하면 하드웨어를 FPGA로 에뮬레이션 했을 때 많은 성능 감소가 발생함을 알 수 있다.

본 논문에서는 설계를 마친 하드웨어를 FPGA 디바이스로 에뮬레이션 했을 때 HDL 코딩 방법에 따른 FPGA에서의 성능 차이가 얼마나 나는지 실험하고 평가하였다. 실험을 위해서는 동일한 기능을 가진 하드웨어를 여러 가지 방법으로 코딩을 해야 하는데 본 연구에서는 가산기[1]를 구현하는 여러 가지 알고리즘을 사용하였다. 실험에서 구현한 가산기는 Ripple Carry 가산기, Carry Lookahead 가산기, 그리고 Prefix 가산기 5종류이다.

2. 관련 연구

Xing[2]은 여러 가지 가산기들을 FPGA 디바이스로 에뮬레이션 하여 성능을 평가하였다. 실험 결과 가산기의 성능은 낮은 비트수에서는 Carry Select, Ripple Carry, Carry Skip 가산기 순으로 나타났지만 비트수가 증가함에 따라 Ripple Carry 가산기의 성능이 낮아진다는 사실을 확인하였다. 이것은 Carry Skip 가산기가 Ripple Carry 가산기에 비해 성능이 앞서고, Carry Select 가산기가 Ripple Carry 가산기에 비해 비용 면에서는 불리하지만 속도 면에서는 많은 향상을 가져 온다는 사실을 확인할 수 있는 결과였다. 그러나 실험에 사용된 가산기의 종류가 다양하지 못하였고 FPGA 디바이스가 Xilinx의 XC4000

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2011-0003422)

계열로서 최신 디바이스를 반영하지 못하였다. 본 연구에서는 Prefix 가산기의 사용과 함께 디바이스를 제조사별로 저비용 디바이스부터 고성능 디바이스까지 다양하게 적용시켰다는 점에서 Xing의 연구와 차이가 있다.

3. 실험 환경 및 방법

3.1 FPGA 디바이스

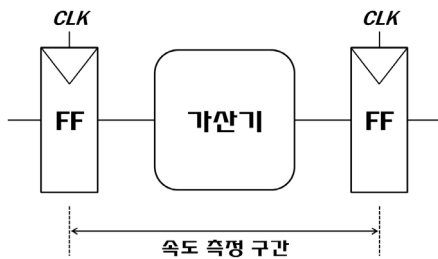
본 연구에서는 <표 1>과 같이 공정 기술에서 차이가 있는 Altera[4]와 Xilinx[5]의 FPGA 디바이스들을 성능별로 3종류씩 총 6종류의 디바이스를 선별하여 실험을 하였다. Altera와 Xilinx의 디바이스를 선택한 이유는 두 회사가 FPGA 시장의 80% 이상을 점유하고 있기 때문이다.

<표 1> 실험에 사용된 Altera와 Xilinx의 FPGA

Altera		Xilinx	
디바이스	공정	디바이스	공정
Arria II GZ	40nm	Kintex 7	28nm
Stratix III	60nm	Virtex 5	65nm
Cyclone II	90nm	Spartan 3E	90nm

3.2 HDL 코딩

디지털 디자인에서 주로 사용되는 HDL에는 VHDL과 Verilog-HDL이 있는데 본 연구에서는 Verilog-HDL을 사용하여 가산기를 코딩하였다. 그리고 조합 논리 회로로 디자인된 가산기의 연산 속도, 즉 입력 신호가 발생한 시점에서 출력신호가 발생한 시점까지의 시간을 측정하기 위해 (그림 1)과 같이 가산기의 입력과 출력 신호에 연결되는 플립플롭을 추가하였다.



(그림 1) 플립플롭을 연결한 가산기

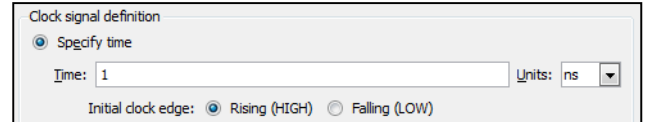
3.3 EDA 툴

FPGA 디바이스로 에플레이션을 하기 위해서는 HDL을 사용한 하드웨어 설계, 오류 수정을 위한 시뮬레이션, HDL 코드를 참조하여 논리 게이트를 생성하는 로직 합성, 생성된 논리 게이트를 해당 디바이스 자원에 배치하고 연결하는 P&R(Place & Route), FPGA 디바이스로의 다운로드 과정 등을 거치게 된다. 본 연구에서는 합성, P&R, 그리고 다운로드를 위해서 Altera의 Quartus II와 Xilinx의 ISE를 사용하였고 시뮬레이션을 위해서 Mentor Graphics의 ModelSim을 사용하였다.

3.4 실행 시간 제약

에플레이션을 할 때 FPGA 디바이스가 높은 성능을 낼 수 있도록 하기 위해서는 합성과 P&R 과정에서 실행 시간에 대한 제약을 강제로 설정할 필요가 있다. 본 연구에서는 (그림 2)와 같이 초기 회로의 지연 시간을 1ns로 설정

했을 후 합성과 P&R 과정을 수행하여 나온 속도를 기준으로 다시 한 번 지연 시간을 설정하고 합성과 P&R 과정을 수행하였다. 두 번째 합성과 P&R 과정의 설정 값은 첫 번째 수행의 결과로 나온 값보다 크지 않은 최대 정수 값을 입력하여 해당 디바이스가 낼 수 있는 최고의 속도를 측정하고자 하였다.



(그림 2) Xilinx ISE의 실행 시간 제약 설정

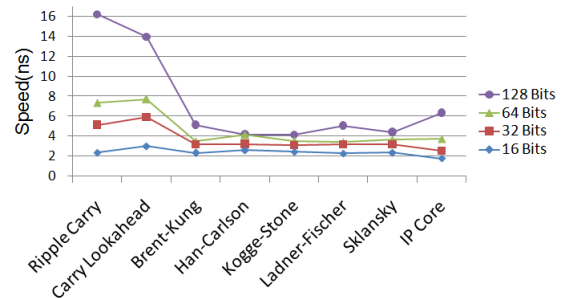
4. 실험 결과 및 분석

4.1 Altera 디바이스

HDL 코딩 방법에 따른 가산기들의 Altera 디바이스에 의한 에플레이션 결과는 <표 2>와 같다.

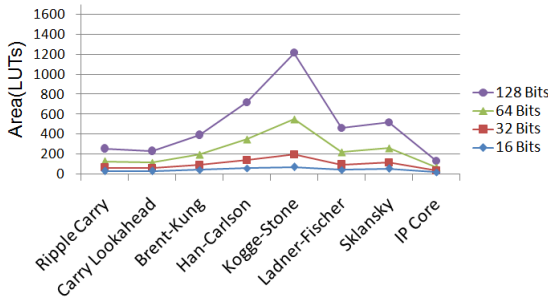
<표 2> Altera 디바이스의 가산기 성능

비트	가산기	Cyclone II		Stratix III		Arria II GZ	
		Speed (ns)	Area (LUTs)	Speed (ns)	Area (LUTs)	Speed (ns)	Area (LUTs)
16	Ripple Carry	6.60	38	2.37	31	1.97	31
	Carry Lookahead	6.38	36	3.01	29	2.00	29
	Brent-Kung	4.29	52	2.30	43	2.05	43
	Han-Carlson	4.06	59	2.61	60	2.46	60
	Kogge-Stone	4.12	77	2.45	70	2.15	70
	Ladner-Fischer	4.26	53	2.26	44	2.33	44
	Sklansky	3.85	57	2.35	54	2.01	54
	IP Core	2.80	18	1.77	18	1.31	18
32	Ripple Carry	12.63	79	5.13	64	3.98	64
	Carry Lookahead	11.98	73	5.89	58	4.15	58
	Brent-Kung	5.36	107	3.18	89	2.29	89
	Han-Carlson	5.26	147	3.20	138	2.62	136
	Kogge-Stone	5.10	208	3.10	195	2.50	194
	Ladner-Fischer	5.28	115	3.18	95	2.36	95
	Sklansky	5.86	132	3.17	115	2.37	116
	IP Core	4.11	34	2.53	34	1.69	34
64	Ripple Carry	22.83	162	7.33	128	6.91	128
	Carry Lookahead	22.52	146	7.71	116	6.11	116
	Brent-Kung	7.31	221	3.48	195	3.24	197
	Han-Carlson	6.15	341	4.14	347	3.35	348
	Kogge-Stone	5.80	569	3.53	548	3.40	525
	Ladner-Fischer	6.53	252	3.40	218	3.21	217
	Sklansky	7.36	280	3.65	258	3.15	257
	IP Core	6.69	66	3.75	66	2.25	66
128	Ripple Carry	45.79	331	16.22	255	13.63	255
	Carry Lookahead	43.65	295	13.96	232	12.98	231
	Brent-Kung	8.58	451	5.13	392	4.00	392
	Han-Carlson	7.10	809	4.15	719	4.03	767
	Kogge-Stone	7.59	1348	4.13	1216	4.04	1203
	Ladner-Fischer	8.74	531	5.04	462	3.96	463
	Sklansky	8.82	551	4.42	516	4.03	515
	IP Core	11.94	130	6.33	130	3.50	130



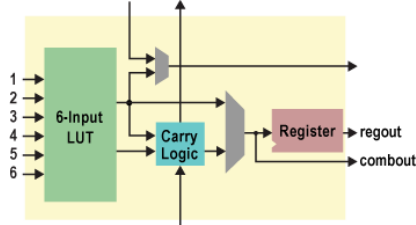
(그림 3) Stratix III의 Speed

서의 에뮬레이션 결과는 <표 4>와 같다.



(그림 4) Stratix III의 Area

- **Speed** : 비트수가 증가할수록 Ripple Carry, Carry Lookahead 가산기와 Prefix 가산기 간의 속도 차이가 많이 발생했고, Prefix 가산기들의 속도 차이는 크지 않았다. 가산기의 일반적인 특성이 잘 반영된 결과이다. <표 3>의 로직 레벨 항목은 속도에 영향을 주는 가산기의 연산 단계를 의미하는데, 로직 레벨을 고려했을 때 Kogge-Stone과 Sklansky 가산기의 속도는 비슷해야 하고 Brent-Kung 가산기보다는 2배 정도 빠른 속도로 동작해야 하지만 디바이스나 비트수에 따라 다른 결과를 보이기도 하였다. 이것은 FPGA의 구조적 특성으로 인해 특정 가산기에 최적화된 논리 회로를 구성하지 못하기 때문으로 판단된다.
- **Area** : FPGA에서 조합 논리 회로를 구성하는 최소 단위는 LUT(Look Up Table)이다. LUT의 사용은 Area, 즉 하드웨어 비용을 결정하게 된다. <표 3>의 셀 항목은 Generate / Propagate 연산과 같이 LUT의 사용에 영향을 주는 가산기의 주요 연산 횟수를 의미한다. 셀 수를 고려했을 때 Brent-Kung 가산기가 Ripple Carry 가산기보다 2배의 비용이 들고 Kogge-Stone 가산기가 Sklansky 가산기보다 2배의 비용이 든다는 사실을 Altera 디바이스는 잘 보여 주었다.



(그림 5) Virtex 5의 LUT와 FF

- **IP Core** : 디지털 디자인의 효율성과 성능 향상을 위해 제조사가 디바이스 내에 미리 정의해 놓은 기능을 IP(Intellectual Property) Core라 한다. Altera의 IP Core를 사용한 가산기는 속도 면에서 Cyclone II, Stratix III의 64비트와 128비트를 제외하면 다른 가산기들보다 우수한 성능을 보였고, 비용 면에서는 디바이스와 비트수가 같으면 최고의 성능을 나타냈다.

<표 3> N비트 가산기의 로직 레벨과 셀

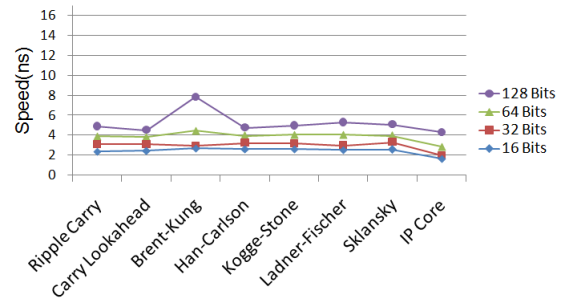
가산기	로직 레벨(Logic Levels)	셀(Cells)
Ripple Carry	N-1	N
Brent-Kung	$2\log_2 N - 1$	2N
Kogge-Stone	$\log_2 N$	$N\log_2 N$
Sklansky	$\log_2 N$	$0.5N\log_2 N$

4.2 Xilinx 디바이스

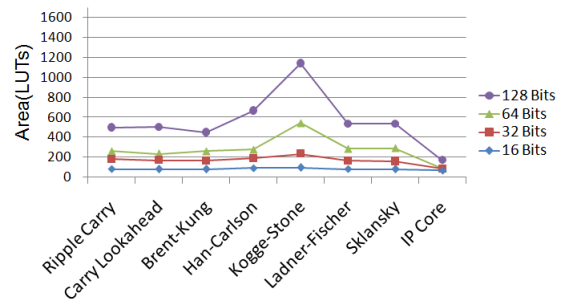
HDL 코딩 방법에 따른 가산기들의 Xilinx 디바이스에

<표 4> Xilinx 디바이스의 가산기 성능

비트	가산기	Spartan 3E		Virtex 5		Kintex 7	
		Speed (ns)	Area (LUTs)	Speed (ns)	Area (LUTs)	Speed (ns)	Area (LUTs)
16	Ripple Carry	5.05	106	2.34	80	1.76	75
	Carry Lookahead	5.12	105	2.42	79	1.79	72
	Brent-Kung	5.22	112	2.68	79	1.75	76
	Han-Carlson	6.03	110	2.59	91	1.84	82
	Kogge-Stone	5.45	130	2.59	95	1.71	86
	Ladner-Fischer	5.13	106	2.51	80	1.89	72
	Sklansky	5.20	117	2.54	78	1.94	82
	IP Core	3.44	65	1.65	65	1.31	65
32	Ripple Carry	7.04	205	3.09	181	2.25	129
	Carry Lookahead	6.74	205	3.07	169	2.26	125
	Brent-Kung	6.33	213	2.92	163	2.85	139
	Han-Carlson	6.98	255	3.22	188	2.14	126
	Kogge-Stone	6.99	339	3.17	233	2.30	210
	Ladner-Fischer	6.98	233	2.96	165	2.27	141
	Sklansky	6.95	246	3.30	159	2.27	107
	IP Core	4.21	81	1.96	81	1.36	64
64	Ripple Carry	8.16	365	3.88	259	2.66	244
	Carry Lookahead	8.41	365	3.80	230	2.81	226
	Brent-Kung	7.98	401	4.47	259	2.59	268
	Han-Carlson	8.02	467	3.93	275	2.80	316
	Kogge-Stone	8.10	711	4.06	543	2.92	443
	Ladner-Fischer	8.91	356	4.06	285	3.61	252
	Sklansky	8.97	396	3.91	289	2.72	294
	IP Core	5.96	101	2.86	85	1.82	181
128	Ripple Carry	10.47	728	4.89	496	3.24	470
	Carry Lookahead	9.98	728	4.51	502	3.25	520
	Brent-Kung	10.00	848	7.84	448	3.59	569
	Han-Carlson	9.14	982	4.74	667	3.23	695
	Kogge-Stone	9.98	1547	4.96	1142	3.76	1138
	Ladner-Fischer	11.72	716	5.29	534	3.12	621
	Sklansky	11.95	731	5.08	533	3.66	578
	IP Core	9.83	201	4.31	169	2.72	371



(그림 6) Virtex 5의 Speed



(그림 7) Virtex 5의 Area

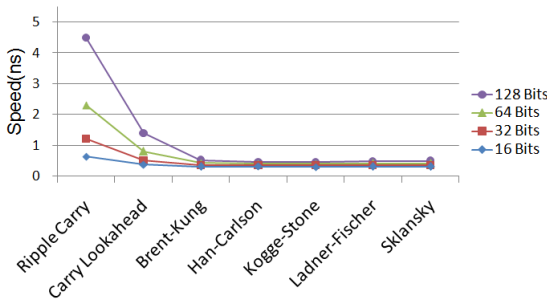
- **Speed** : Ripple Carry, Carry Lookahead 가산기와 Prefix 가산기 간의 속도 차이가 크지 않았고 일부의 Prefix 가산기는 Ripple Carry, Carry Lookahead 가산기보다 느린 속도를 보였다. 이 같은 결과는 FPGA

의 내부 구조가 프로그래밍이 가능한 복잡한 논리 블록과 내부 연결선으로 이루어져 있기 때문에 HDL 코딩 방법에 따른 성능 특성이 디바이스에 잘 반영되지 못한 것으로 판단된다. Prefix 가산기 중에서는 Ladner-Fischer와 Sklansky 가산기가 비교적 낮은 성능의 디바이스인 Spartan 3E의 64비트 이상에서 속도가 느려지는 경향을 보여 <표 3>에 나타난 로직 레벨을 잘 반영하지 못하였다. 실험 결과를 통해 디바이스에 따라서는 Prefix 가산기를 사용하더라도 좋은 성능을 보이지 않을 수 있다는 것을 확인하였다.

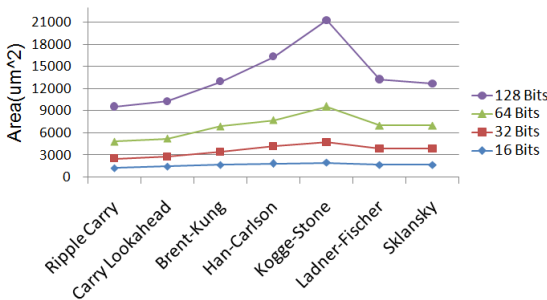
- **Area** : <표 3>에 나타난 셀 수를 고려했을 때 Ripple Carry 가산기가 Brent-Kung 가산기에 비해 2배 정도 우수한 결과를 보여야 하지만 그렇지 않았으며 오히려 Virtex 5 디바이스에서는 Brent-Kung 가산기의 성능이 조금 우수하게 나왔다. 그러나 대체적으로 Altera 디바이스나 ASIC 디바이스와 마찬가지로 코딩 방법에 따른 높은 성능 민감도를 보였다.
- **IP Core** : Xilinx에서 제공하는 IP Core를 사용한 가산기의 경우 속도 면에서 Spartan 3E의 128비트를 제외하면 다른 가산기들보다 우수한 성능을 보였고, 비용 면에서는 Altera 디바이스와 마찬가지로 디바이스와 비트수가 같으면 항상 우수한 성능을 보여 주었다.

4.3 ASIC 디바이스

TSMC 90nm[6] 공정 기술로 제조된 ASIC 디바이스는 HDL 코딩 방법에 따라 (그림 8), (그림 9)과 같은 성능 그래프를 보여 주었다.



(그림 8) ASIC(TSMC 90nm)의 Speed



(그림 9) ASIC(TSMC 90nm)의 Area

- **Speed** : Ripple Carry, Carry Lookahead 가산기와 Prefix 가산기 간의 속도 차이는 뚜렷하게 나타났으나, Prefix 가산기들의 속도 차이는 크지 않았다. FPGA에서는 IP Core를 사용한 가산기들이 상대적으로 우수한 성능을 보여 주었는데 ASIC(TSMC 90nm)은 FPGA의 IP Core를 사용한 가산기보다 약 4배 정

도의 속도 향상을 가져왔다. ASIC은 속도 향상을 위하여 비용 부담이 있더라도 고속의 Prefix 가산기를 사용하는 것이 효과적이라는 것을 확인 시켜주는 결과였다.

- **Area** : Ripple Carry, Carry Lookahead, 그리고 Prefix 가산기 순으로 비용이 증가하였고, Prefix 가산기들도 많은 차이를 보이면서 비용 증감이 일어났다. 이것은 ASIC에서 HDL 코딩 방법에 따른 가산기의 비용 민감도가 크다는 것을 말해준다.

5. 결론

본 논문에서는 설계된 하드웨어의 검증을 위해 사용되는 FPGA 디바이스들이 HDL 코딩 방법에 따라 어느 정도의 성능 민감도를 가지는지 속도와 비용 면에서 실험하고 평가하였다. 본 연구를 통해 HDL 코딩 방법이 다르면 제조사별, 디바이스별로 성능 차이가 날 수 있음을 확인하였다. 속도 면에서 가산기의 일반적인 성능 특성이 Altera 디바이스에서는 잘 반영되고 있는 반면 Xilinx 디바이스에서는 그렇지 못하였다. 비용 면에서는 Altera 디바이스와 Xilinx 디바이스가 비슷한 경향을 보였으나 ASIC과 좀 더 가까운 성능 민감도를 보인 것은 Altera 디바이스였다. 이러한 결과는 Altera 디바이스가 Xilinx 디바이스보다 HDL 코딩 방법에 따른 성능 민감도가 높으며 상대적으로 Xilinx 디바이스의 경우 HDL 코딩 방법에 영향을 적게 받는다는 것을 의미한다. 제조사에서 제공하는 IP Core의 성능 또한 확인할 수 있었는데 성능이 낮은 디바이스의 높은 비트수를 제외하면 속도 면에서 다른 가산기들보다 우수한 성능을 보였고, 비용 면에서는 디바이스와 비트수가 같으면 항상 최고의 성능을 보여 주었다. 성능 비교의 기준이 될 수 있는 ASIC의 경우는 속도 면에서 IP Core를 사용한 가산기보다 4배 정도의 우수한 성능을 보였다.

본 연구를 바탕으로 향후에는 제조사와 디바이스에 따라 차이를 보이는 성능 민감도의 정확한 분석을 위해 FPGA 디바이스의 내부 구조에 대하여 연구할 필요가 있으며, 또한 CPU와 같이 복잡하고 고성능을 요구하는 하드웨어에 대한 실험도 진행하여 더욱 효과적인 FPGA 에뮬레이션 방법을 찾을 필요가 있다고 생각된다.

참고문헌

- [1] Israel Koren, "Computer Arithmetic Algorithms, 2nd Edition", A K Peters, Ltd., 2002
- [2] Shanzhen Xing, William W.H. Yu. "FPGA Adders: Performance Evaluation and Optimal Design". IEEE Design & Test of Computer, January-march 1998
- [3] Graham Schelle, Jamison Collins, Ethan Schuchman, Perry Wang, Xiang Zou, Gautham Chinya, Ralf Plate, Thorsten Mattner, Franz Olbrich, Per Hammarlund, Ronak Singhal, Jim Brayton, Sebastian Steibl, Hong Wang. Intel® Nehalem Processor Core Made FPGA Synthesizable. In FPGA '10
- [4] Altera Corporation, "Quartus II 9.1 Web Edition", Available: <http://www.altera.com>
- [5] Xilinx Corporation, "Xilinx ISE Design Suite 13.1", Available: <http://www.xilinx.com>
- [6] Taiwan Semiconductor Manufacturing Company, "90 nm Technology", Available: <http://www.tsmc.com/english/dedicatedFoundry/technology/90nm.htm>