

논문 2008-45SD-8-9

다중버스 아키텍처 구조에서 고성능 DMA를 이용한 TFT-LCD Controller 구현

(TFT-LCD Controller Implementation Using DMA of High Performance in Multi-Bus Architecture)

이국표*, 이근환**, 윤영섭*

(Kook-Pyo Lee, Keun-Hwan Lee, and Yung-Sup Yoon)

요약

버스 아키텍처는 통신 전송을 만드는 마스터, 전송을 응답받는 슬레이브, 마스터를 선택하는 아비터, 그리고 버스를 연결해주는 브리지 등으로 구성되어 있다. 이것은 최근에 좀 더 복잡해지고 있으며, 다중버스 아키텍처로 발전하고 있다. 본 논문에서는 여러 다중 shared bus 구조에 대해 논의해 보고 브리지의 레이턴시를 줄이기 위해서 메모리 선택터를 도입한 구조를 제안하였다. 마지막으로 이 버스구조에 DMA 마스터를 사용하는 LCD 컨트롤러를 집적하였으며, RTL 시뮬레이션과 FPGA 보드 테스트를 통하여 검증하였다. ModelSim 툴을 이용한 타이밍 시뮬레이션에서 DMA, LCD 라인버퍼, SDRAM 컨트롤러 등이 정상적으로 동작되었으며, LCD 패널이 장착된 실제 FPGA 보드에서 LCD 이미지를 확인하였다.

Abstract

The bus architecture consists of a master initiating a communication transaction, a slave responding to the transaction, an arbiter selecting a master, a bridge connecting buses and so on. Recently this is more complicated and developed toward multi-bus architecture. In this paper, several cases of multi-shared bus architecture are discussed and in order to decrease the bridge latency, the architecture introducing a memory selector is proposed. Finally, a LCD controller using DMA master is integrated in this bus architecture that is verified due to RTL simulation and FPGA board test. DMA, LCD line buffer and SDRAM controller are normally operated in the timing simulation using ModelSim tool, and the LCD image is confirmed in the real FPGA board containing LCD panel.

Keywords : TFT-LCD controller, DMA, FPGA, bus architecture

I. 서론

단일 칩으로 여러 기능을 동시에 사용하고자 하는 사용자의 요구를 충족시켜주기 위해, 여러 프로세서, 각종 용도의 DMA, DSP 등을 하나의 칩에 구현하려는 노력이 지속적으로 진행되고 있다^[1~2]. 데이터 전송을 실행하는 마스터, 데이터를 받는 슬레이브, 버스중재 장치인 아비터, 슬레이브를 결정하는 디코더, 버스간의 연결부

인 브리지 등으로 구성된 버스 아키텍처는 좀 더 복잡해지고 있으며, 2-3단의 다중버스 아키텍처 구조로 발전하고 있다^[3~4]. 다중버스 아키텍처에서 마스터 간의 버스점유 요청과 버스 간의 메모리 접근 요청 등이 동시에 발생할 경우 버스 성능이 크게 떨어질 수 있다. 그래서 AMBA 프로토콜을 이용하는 다중버스 구조에서는 마스터를 포함하는 AHB 버스와 마스터를 포함하지 않는 APB 버스가 결합되어 있어서, 메모리 접근 요청을 AHB 버스만 하도록 간단하게 구성하였다^[5].

그러나 사용자의 다양한 요구의 충족과 여러 시스템이 단일 SoC 칩으로 집적화되는 발전 속도에 따라, 마스터의 개수와 종류가 다양해 질 수밖에 없다. 결국

* 정회원, ** 학생회원, 인하대학교 전자공학과

(Dept. of Electronics Engineering, Inha University)

※ 이 논문은 인하대학교의 지원에 의하여 발간되었음.

접수일자: 2008년3월21일, 수정완료일: 2008년7월14일

다중버스에 여러 마스터들이 구성되어지게 되며, 복잡도는 크게 증대하게 된다^[3~4].

본 연구에서는 버스 아키텍처의 구조에 따른 장단점을 분석해 보고, 3단 이상의 다중버스 아키텍처의 구성 방법에 대해 논하려고 한다. 그리고 VHDL을 이용하여 ARM922T 프로세서, DMA(Direct Memory Access), SDRAM, SRAM, TFT-LCD 컨트롤러 등으로 구성된 3단 다중버스 아키텍처를 설계해 보고, 특징을 분석하려고 한다.

마지막으로 다중버스 아키텍처에 고성능 DMA와 라인버퍼를 내장한 TFT-LCD 컨트롤러를 설계 및 합성하였다. 그리고 본 설계 DB를 프로그래밍한 FPGA 보드에서 LCD 패널에 이미지가 정상적으로 전송됨을 확인하였다.

II. 다양한 버스 아키텍처 구조

버스 아키텍처의 구조는 사용자의 용도와 성능향상 측면에서 발달해 오고 있다^[3~4]. 그림 1(a)는 다중포트를 지원하는 메모리를 이용하여, 각 마스터 컴포넌트 C1, C2, C3이 메모리를 사용하는 구조이다. 3개의 컴포넌트가 메모리를 사용할 수 있도록 인터페이스를 별도로 제작한 구조로서, 메모리에 동시에 접근할 수 있는 이상적인 구조이다. 그림 1(b)는 일반적으로 많이 사용되고 있는 공용 버스 아키텍처이다. 버스1과 버스2에 싱글포트를 지원하는 메모리가 있는 구조로서, 여러 컴포넌트가 동시에 메모리에 접근할 경우 아비터1과 아비터2의 중재에 의해 메모리를 사용할 수 있다. 우선순위에 의해 메모리접근이 이루어지므로, 그림 1(a)의 구조에 비하여 메모리접근을 위해 기다리는 시간만큼 성능이 저하될 수 있다. 버스1과 버스2는 각각 독립적인 컴포넌트와 메모리를 구성할 수 있으며, 버스1과 버스2는 브리지에 의해 연결될 수 있다. 만약 버스1의 컴포넌트 C1이 버스2의 메모리2를 접근하고자 한다면, 반드시 브리지를 경유해야 한다. 그림1(c)는 분리형 버스 아키텍처이며, 버스와 메모리를 분리시켜서 컴포넌트 C1이 메모리1을 사용하고, 컴포넌트 C2가 메모리2를 사용하고, 컴포넌트 C3이 메모리3을 사용함에 따라, 여러 컴포넌트가 동시에 메모리에 접근할 때 발생하는 지연시간을 줄일 수 있다.

그러나 버스를 1/3로 분리하였으므로 각 컴포넌트가

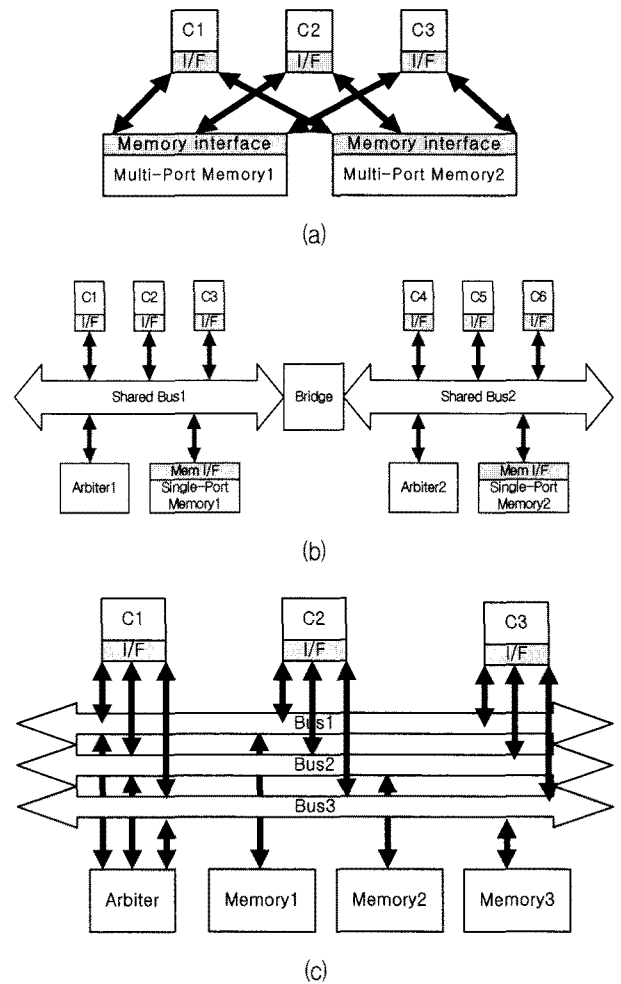


그림 1. (a) 이상적인 버스 아키텍처, (b) 공용버스 아키텍처, (c) 분리형 버스 아키텍처

Fig. 1. (a) Ideal bus architecture, (b) shared bus architecture, (c) split bus architecture

한 번에 처리하는 데이터양 또한 1/3로 줄어들어서 버스 bandwidth가 감소된다. 버스 Bandwidth는 수식 (1)과 같이 주파수 f 와 버스에서 한 번에 처리하는 바이트 수 N_{byte} 의 곱으로 표현된다.

$$B = f \cdot N_{byte} \tag{1}$$

결국, 분리형 버스 아키텍처는 아비터에 의한 웨이팅 시간을 감소시킬 수 있지만 버스 bandwidth가 감소되는 특성을 가지므로, 버스 bandwidth 보다 아비터에 의한 웨이팅시간에 의해 버스 성능이 결정되는 구조에 사용될 수 있다.

III. 다중버스 아키텍처 구성

본 연구에서는 브리지 통과에 의한 성능저하를 개선

하기 위하여 그림 1(b)의 공용버스 아키텍처에 메모리 셀렉터를 도입하여 그림 2에 나타내었다.

버스 아키텍처에서 브리지를 통과할 때 발생하는 레이턴시를 줄일 수 있으며, 버스1과 버스2에서 동시에 메모리를 사용할 경우, 메모리 셀렉터에 의해 중재된다. 그림 2에 표현된 것처럼 본 연구의 버스 아키텍처는 AHB1 버스에 ARM922T 프로세서, watchdog 타이머, 인터럽트 컨트롤러 등을 구성하였으며, AHB2 버스에는 고속을 요구하지 않는 PLL, 타이머, UART, 외부버스 인터페이스 EBI, FPGA 인터페이스 등을 구성하였다. 그리고 AHB1, AHB2 버스에 대해서 메모리 셀렉터에 의해 공용 SDRAM, SRAM 메모리가 조절되고 있다.

FPGA 인터페이스 블록에 의해서 추가적으로 버스를 구성할 수 있는데, 이는 그림 3에 나타나 있다. 마스터 컴포넌트 C1, C2, C3을 구성할 수 있으며, 메모리, 슬레이브 블록, 아비터 블록도 추가할 수 있다.

SoC 버스 아키텍처에서 필수적으로 사용하는 기본 블록들은 AHB1, AHB2 버스를 통해 구성하고, 추가적인 버스구조를 통해 새로운 블록을 추가할 수 있도록 해서, 버스구조를 조직적으로 만들었을 뿐만 아니라,

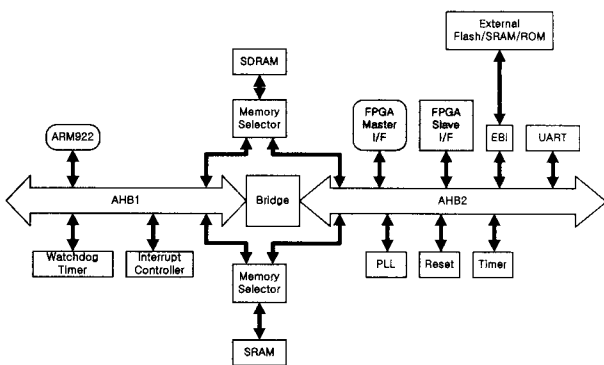


그림 2. 수정된 다중 AHB 버스 아키텍처
Fig. 2. Modified Multi-AHB bus architectures.

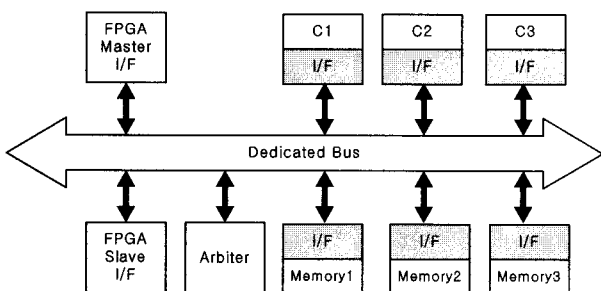


그림 3. 사용자가 추가로 설계할 마스터와 슬레이브를 위한 전용 버스 아키텍처
Fig. 3. Dedicated bus architecture for masters and slaves that will be designed additionally.

FPGA를 통해 손쉽게 조절 가능한 버스구조를 확보하였다.

IV. 전용 DMA를 이용한 TFT-LCD 구현

III 장의 다중 AHB 버스 아키텍처에 전용 DMA를 장착한 TFT-LCD 컨트롤러를 설계하였다. 그림 4에 전형적인 LCD 컨트롤러를 사용한 TFT-LCD 구동방법이 나타나 있는데, 동작 순서는 표 1에 나타내었다. 그리고 DMA를 내장한 TFT-LCD 컨트롤러를 사용한 TFT-LCD 구동방법은 그림 5와 표 2에 나타내었다.

위의 두 방법에 대한 특징적인 차이는 TFT-LCD 영상데이터를 전송하는 주체이다. 프로세서가 직접 데이터를 전송할 수도 있으며, DMA를 통해 독자적으로 전송하는 것도 가능하다. 예전에는 TFT-LCD 구동하는 칩의 프로세서는 TFT-LCD만 담당하는 형태라서, 프로세서가 직접 영상데이터를 전송하였다. 그러나 최근의 추세는 하나의 프로세서가 여러 가지 구성요소를 모두 제어하는 SoC 칩 형태로 발전하고 있다. 결국 프로세서는 전체적인 시스템을 운영해야 하며, TFT-LCD 구동하는데 전담할 수 없게 되어 DMA 마스터 등을 통해 업무를 분담시켜야 한다.

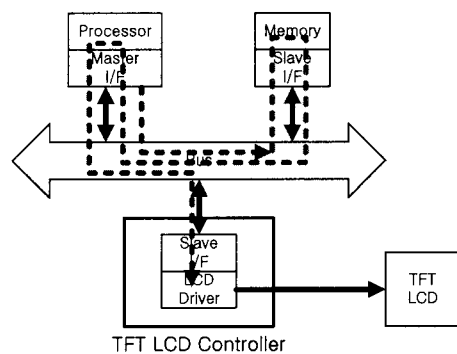


그림 4. 전형적인 TFT-LCD 컨트롤러를 사용한 TFT-LCD 구동방법
Fig. 4. The driving method of TFT-LCD using conventional TFT-LCD controller.

표 1. 전형적인 TFT-LCD 구동방법
Table 1. The conventional driving method of TFT-LCD.

순서	내용
1	프로세서가 LCD 영상 데이터를 메모리로부터 읽기
2	데이터를 프로세서 레지스터에 저장
3	저장된 레지스터 데이터를 TFT-LCD 컨트롤러에 전달
4	TFT-LCD 컨트롤러의 데이터를 TFT-LCD 패널에 전달

표 2. DMA를 사용한 TFT-LCD 구동방법
Table 2. The driving method of TFT-LCD using DMA.

순서	내용
1	프로세서가 TFT-LCD 컨트롤러의 "Enable" 레지스터에 값을 기입하여 TFT-LCD를 동작시킴
2	TFT-LCD 컨트롤러의 레지스터에 저장된 명령에 맞게 DMA가 메모리와 통신하며 데이터를 라인버퍼에 저장
3	라인버퍼 데이터가 LCD 드라이버를 거쳐서 TFT-LCD 패널에 전달

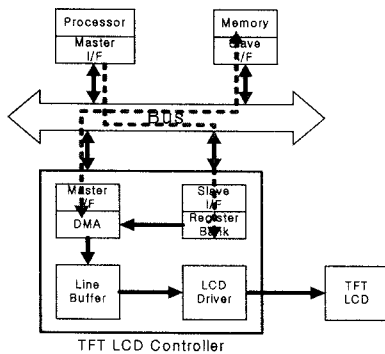
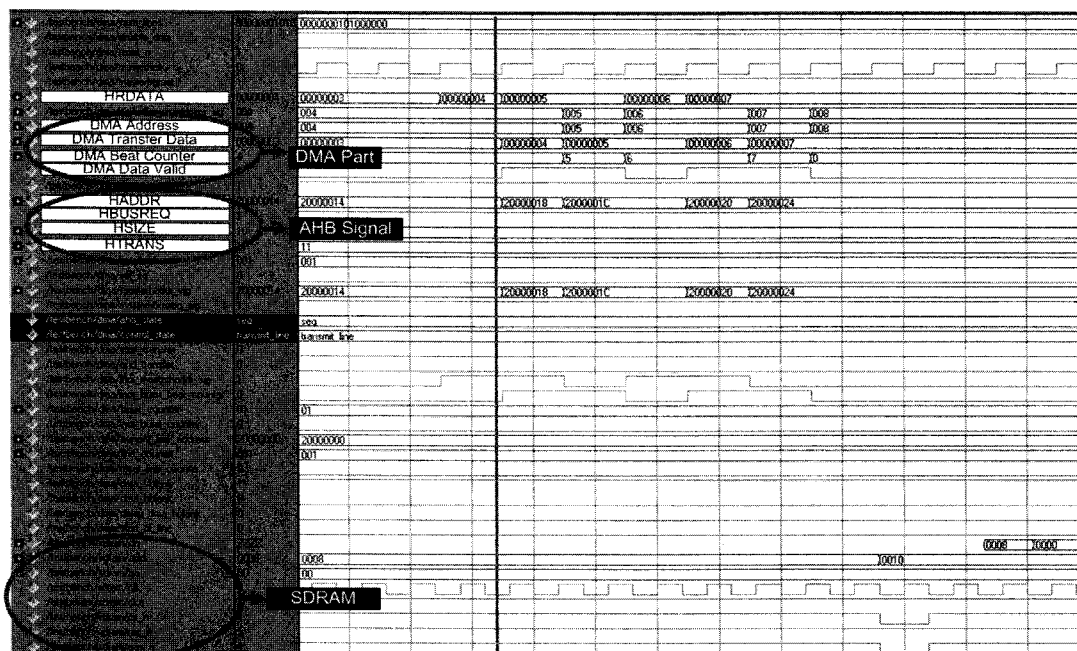


그림 5. DMA 내장형 TFT-LCD 컨트롤러를 사용한 TFT-LCD 구동방법
Fig. 5. The driving method of TFT-LCD using DMA embedded TFT-LCD controller.

만약 여러 마스터와 슬레이브가 통합되어 있는 SoC 칩의 프로세서가 TFT-LCD 구동을 직접 제어할 경우, 다른 구성요소의 성능이 떨어질 뿐만 아니라, TFT-LCD 영상데이터 또한 원하는 속도에 전달받지 못하여 성능에 큰 제약이 된다. 그림 6과 그림 7은 설계된 DMA 내장 TFT-LCD 컨트롤러의 시뮬레이션 결과이다. 회로는 VHDL로 설계하였으며 시뮬레이션은 "ModelSim SE 6.2b" 툴을 사용하였다^[6]. 그리고 "ARM ADS1.2"의 ARM922T 모델과 구성요소를 "Altera Quartus II" 툴로 통합하였다^[7]. 세부적인 설계블록의 계층도는 그림 8에 나타내었다.

그림 6에 보듯이 DMA 블록에서 발생하는 AHB 신호에 의해 SDRAM 메모리 데이터가 TFT-LCD 컨트롤러의 "DMA Transfer Data" 레지스터에 저장됨을 확인할 수 있다. "DMA Transfer Data" 레지스터에 저장된 데이터는 라인버퍼를 통하여 외부로 출력되는데, 그림 7에 출력 파형이 나타나 있다. 여기서 "bled" 신호는 back light enable/disable 신호이고, "lcd_de" 신호는 LCD data enable 신호이며 "lcd_data" 신호는 16비트 LCD 데이터 신호이고, "lcd_mclk" 신호는 LCD 클럭신호이다. 위의 신호가 그림 9의 LCC3600A 타이밍 컨트롤 칩에 전달되어 TFT-LCD가



HRDATA: AHB read data, HADDR: AHB address, HBUSREQ: AHB bus request, HSIZE: AHB data size, HTRANS: AHB transfer signal, DMA Beat Counter: DMA transfer counter, DMA Data Valid: DMA valid transfer signal

그림 6. DMA 블록의 출력파형
Fig. 6. Output waveform of DMA block.

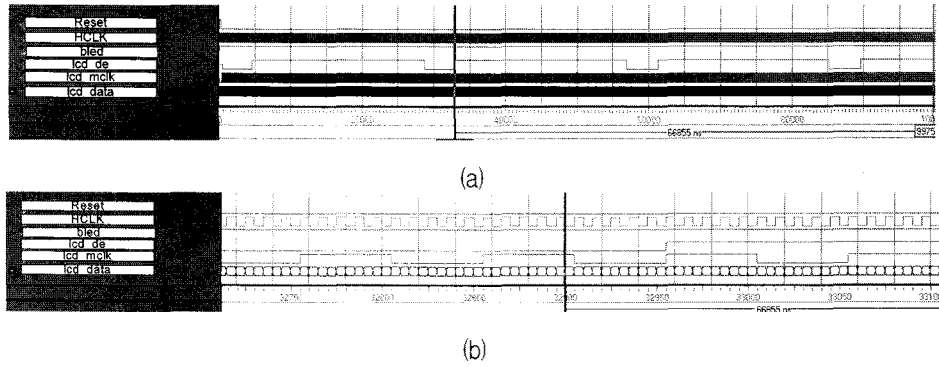


그림 7. LCD 컨트롤러의 출력파형 : (a) 수십 us대의 파형, (b) 수십 ns대의 파형

Fig. 7. Output waveform of LCD controller :

(a) waveform of several tens of micro-second scale, (b) waveform of several tens of nano-second scale.

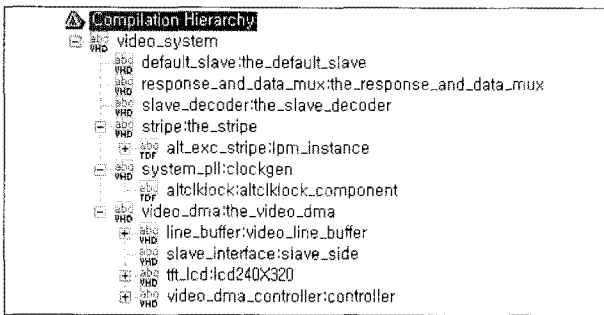


그림 8. 세부적인 설계블록 계층도

Fig. 8. The hierarchy of design block in detail.

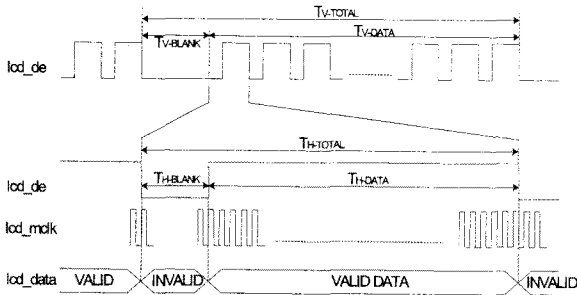


그림 9. LCC3600A 타이밍 컨트롤 칩의 타이밍도

Fig. 9. The timing diagram of Timing Control Chip of LCC3600A.

구동되는데, “lcd_de” 신호가 “high”에서 “lcd_mclk” 클럭신호에 따라 “lcd_data” 신호가 TFT-LCD 패널에 전송된다^[8]. 본 시뮬레이션으로 AHB 프로토콜의 DMA 동작과 TFT-LCD 컨트롤러가 정상 동작됨을 확인할 수 있었다.

V. FPGA결과

VHDL RTL 코드를 FPGA로 구현하였으며, 그림 10의 Excalibur APEX20KE FPGA 칩을 사용하였다. 그

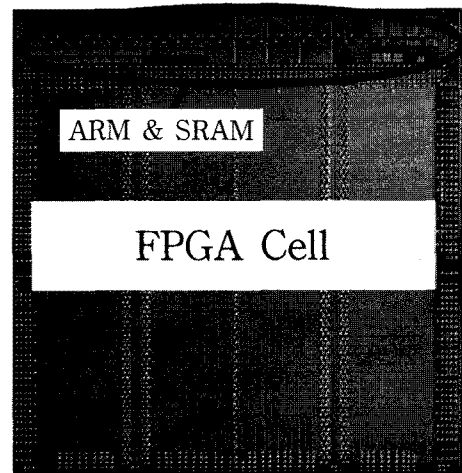


그림 10. Excalibur APEX20KE의 칩사진

Fig. 10. Chip photograph of Excalibur APEX20KE.

표 3. 합성 결과

Table 3. Synthesized results.

합성결과	값
System Frequency	25MHz
PLL Frequency	60MHz
Total logic elements	662
Total pins	127
Total memory bits	16,384
Total PLL	1
Total combinational function	488
Total register	330
Average fan-out	3.91
Max fan-out	352

림 10의 상단부분에 ARM 프로세서와 SRAM 등의 미리 만들어진 블록이 있으며, 나머지 부분에는 FPGA 셀이 존재하게 된다. 위의 칩에 설계 블록을 프로그래밍하기 전에 “Altera Quartus II” 툴을 이용하여 합성하였는데, 그 세부적인 결과는 표 3에 나타내었다. 설계 시

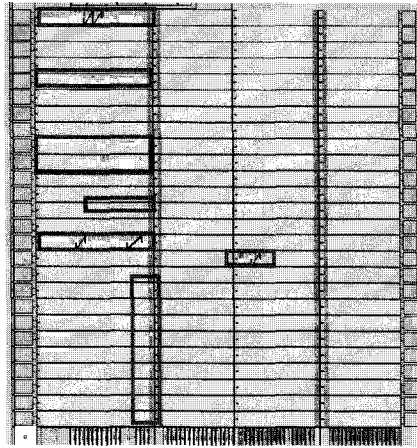


그림 11. 합성 후 floorplan 결과
Fig. 11. Floorplan results after synthesization.

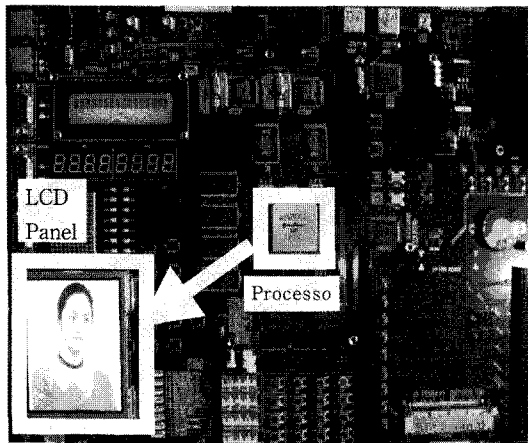


그림 12. 프로그래밍 후 FPGA 보드 동작
Fig. 12. The operation of FPGA board after programming.

비교적 크기가 큰 DMA, 라인버퍼 등을 모두 포함해서 약 16,000 정도의 메모리 비트만으로 합성이 가능하였다. 평균적인 fan-out은 약 4정도이었으며, 최대 fan-out은 약 352정도이었지만 60MHz까지 setup, hold 마진을 충족하며 합성이 가능하였다.

그림 11에는 합성되어 사용된 FPGA 셀을 나타내고 있다. "Altera Quartus II" 툴에 의해서 타이밍 상 최적인 부분의 셀을 사용하고 있으며, 사용된 셀은 네모상자로 표시되어 있다. 그림 11에서는 Excalibur APEX20KE FPGA 칩, LCC3600A 타이밍 컨트롤 칩, LCD 패널이 장착된 보드에서 TFT-LCD의 동작을 나타내고 있다.

위의 FPGA 보드 테스트를 통해서, 멀티 버스구조에서 프로세서와 TFT-LCD가 정상 동작됨을 확인할 수 있었다.

VI. 결 론

많은 컴포넌트를 하나의 칩에 구현하기를 희망하는 사용자의 요구에 따라 단일 shared bus 구조는 다중 shared bus 구조로 급속하게 변화하고 있다. 본 연구에서는 여러 다중 shared bus 구조에 대해 고찰해 보고 브리지의 레이턴시를 줄이기 위해서 메모리 셀렉터를 도입한 구조를 제안하였다. 그리고 위의 버스구조에 DMA 마스터가 장착된 LCD 컨트롤러를 집적시켜서 정상 동작됨을 확인할 수 있었다. VHDL 코드를 이용하여 설계된 DB를 ModelSim 툴로 타이밍 시뮬레이션한 결과에서 AHB 구조 하에서 DMA, LCD 라인버퍼, SDRAM 컨트롤러 등이 잘 동작되었으며, LCD 패널이 장착된 실제 FPGA 보드에서 이미지를 확인하였다.

참 고 문 헌

- [1] R. Lu and C.-K. Koh, "SAMBA-Bus: A High Performance Bus Architecture for System-on-Chips", *IEEE Trans. on VLSI Systems*, vol. 15, no. 1, pp.69-79, 2007.
- [2] M. Anders, N. Rai, R. Krishnamurthy and S. Borkar, "A transition-encoded dynamic bus technique for high-performance interconnects", *IEEE J. Solid-State Circuits*, vol. 5, no. 4, pp.444-454, 1997.
- [3] K. Lahiri, A. Raghunathan, and S. Dey, "Design Space Exploration for Optimizing On-Chip Communication Architectures", *IEEE Trans. Computer-Aided Design*, vol. 23, pp.952-961, June. 2004.
- [4] K. Lahiri, A. Raghunathan, and S. Dey, "System-Level Performance Analysis for Designing On-Chip Communication Architectures", *IEEE Trans. Computer-Aided Design*, vol. 20, no. 6, pp. 768-783, June 2001.
- [5] AMBA TM Specification(AHB) (Rev 2.0), ARM Ltd, May 1999.
- [6] <http://www.model.com>
- [7] <http://www.altera.com>
- [8] <http://alldatasheet.com/datasheet-pdf/pdf/88666/SAMSUNG/LCC3600A.html>

저 자 소 개



이 국 표(학생회원)
 1999년 인하대학교 전자재료
 공학과 학사 졸업.
 2001년 인하대학교 전자재료
 공학과 석사 졸업.
 2005년~현재 인하대학교 전자
 공학과 박사과정
 (박사수료)

<주관심분야 : 반도체, SoC, 디지털, 아날로그 회
 로설계, FPGA설계, 버스 아키텍처 성능분석>



이 근 환(학생회원)
 2007년 순천향대학교 전기전자
 공학과 학사 졸업.
 2007년~현재 인하대학교
 전자공학과 석사과정.
 <주관심분야 : 반도체, SoC, 디지
 털 회로설계>



윤 영 섭(정회원)
 1975년 서울대학교
 금속공학과 학사 졸업.
 1977년 한국과학원
 재료공학과 석사 졸업.
 1988년 Univ. Southern California
 전자공학과 박사 졸업.

1987년~1988년 Oklahoma State University
 대우교수

1988년~1989년 UCLA Device Research Lab.
 연구원

1989년~1992년 삼성전자 기흥반도체연구소
 수석연구원

1992년~현재 인하대학교 전자공학과 교수
 <주관심분야 : ULSI DRAM 을 위한 신물질 개
 발, 강유전성 박막, Pyroelectric 센서, SAW
 device, 회로설계>