

SoC Report

-8051 Core-

과목 : SOC수업

과제명 : 8051 core

담당교수 : 박종성

학과 : 메카트로닉스공학과

학번 : 2015132039

이름 : 최영운

제출일 : 12월 9일

목차

1. 주제 배경이론

- 1) 8051 MCU 구조 이해
- 2) 8051 실습 순서

2. 소스코드 설명

- 1) MCU_Core_component
- 2) 8051 Core & F/W Top Design

3. 시뮬레이션 결과 및 설명

- 8051 Core & F/W 동작확인

4. DE2 실습

- 8051 Core & F/W 동작확인

5. 토의 및 실습소감

Table 1. The MCS[®] 51 Family of Microcontrollers

DEVICE	ROM/EPROM (bytes)	Register RAM (bytes)	Speed (MHz)	I/O Pins	Timer/ Counters	UART	Interrupt Sources	PCA Channels	A/D Channels	SEP	OSC	DMA Channels	Lock Bits	Power Down & Idle Modes
8051 Product Line														
8051AH	ROMLESS	128	12	32	2	1	5	0	0	0	0	0	-	-
8051AH	4K ROM	128	12	32	2	1	5	0	0	0	0	0	0	-
8051AHP	4K ROM	128	12	32	2	1	5	0	0	0	0	0	0	-
8751H	4K EPROM	128	12	32	2	1	5	0	0	0	0	0	1	-
8751BH	4K EPROM	128	12	32	2	1	5	0	0	0	0	0	2	-
8052 Product Line														
8052AH	ROMLESS	256	12	32	3	1	6	0	0	0	0	0	-	-
8052AH	8K ROM	256	12	32	3	1	6	0	0	0	0	0	0	-
8752BH	8K EPROM	256	12	32	3	1	6	0	0	0	0	0	2	-
80C51 Product Line														
80C51BH	ROMLESS	128	12, 16	32	2	1	5	0	0	0	0	0	-	Yes
80C51BH	4K ROM	128	12, 16	32	2	1	5	0	0	0	0	0	0	Yes
80C51BHP	4K ROM	128	12, 16	32	2	1	5	0	0	0	0	0	P	Yes
87C51	4K EPROM	128	12, 16, 20, 24	32	2	1	5	0	0	0	0	0	3	Yes
8XC52/54/58 Product Line														
80C52	ROMLESS	256	12, 16, 20, 24	32	3	1	6	0	0	0	0	0	-	Yes
80C52	8K ROM	256	12, 16, 20, 24	32	3	1	6	0	0	0	0	0	1*	Yes
87C52	8K EPROM	256	12, 16, 20, 24	32	3	1	6	0	0	0	0	0	3	Yes
80C54	16K ROM	256	12, 16, 20, 24	32	3	1	6	0	0	0	0	0	1	Yes
87C54	16K EPROM	256	12, 16, 20, 24	32	3	1	6	0	0	0	0	0	3	Yes
80C58	32K ROM	256	12, 16, 20, 24	32	3	1	6	0	0	0	0	0	1	Yes
87C58	32K EPROM	256	12, 16, 20, 24	32	3	1	6	0	0	0	0	0	3	Yes
8XL52/54/58 Product Line														
80L52	8K ROM	256	12, 16, 20*	32	3	1	6	0	0	0	0	0	1	Yes
87L52	8K OTP ROM	256	12, 16, 20*	32	3	1	6	0	0	0	0	0	3	Yes
80L54	16K ROM	256	12, 16, 20*	32	3	1	6	0	0	0	0	0	1	Yes
87L54	16K OTP ROM	256	12, 16, 20*	32	3	1	6	0	0	0	0	0	3	Yes
80L58	32K ROM	256	12, 16, 20*	32	3	1	6	0	0	0	0	0	1	Yes
87L58	32K OTP ROM	256	12, 16, 20*	32	3	1	6	0	0	0	0	0	3	Yes

Table 1. The MCS[®] 51 Family of Microcontrollers

DEVICE	ROM/EPROM (bytes)	Register RAM (bytes)	Speed (MHz)	I/O Pins	Timer/ Counters	UART	Interrupt Sources	PCA Channels	A/D Channels	SEP	OSC	DMA Channels	Lock Bits	Power Down & Idle Modes
8XC51FA/B/C Product Line														
80C51FA	ROMLESS	256	12, 16	32	3	1	7	5	0	0	0	0	-	Yes
83C51FA	8K ROM	256	12, 16	32	3	1	7	5	0	0	0	0	0	Yes
87C51FA	8K EPROM	256	12, 16, 20, 24	32	3	1	7	5	0	0	0	0	3	Yes
83C51FB Product Line														
83C51FB	16K ROM	256	12, 16, 20, 24	32	3	1	7	5	0	0	0	0	1	Yes
87C51FB	16K EPROM	256	12, 16, 20, 24	32	3	1	7	5	0	0	0	0	3	Yes
83C51FC Product Line														
83C51FC	32K ROM	256	12, 16, 20, 24	32	3	1	7	5	0	0	0	0	1	Yes
87C51FC	32K EPROM	256	12, 16, 20, 24	32	3	1	7	5	0	0	0	0	3	Yes
8XL51FA/B/C Product Line														
80L51FA	ROMLESS	256	12, 16, 20*	32	3	1	7	5	0	0	0	0	-	Yes
83L51FA	8K ROM	256	12, 16, 20*	32	3	1	7	5	0	0	0	0	1	Yes
87L51FA	8K OTP ROM	256	12, 16, 20*	32	3	1	7	5	0	0	0	0	3	Yes
83L51FB	16K ROM	256	12, 16, 20*	32	3	1	7	5	0	0	0	0	1	Yes
87L51FB	16K OTP ROM	256	12, 16, 20*	32	3	1	7	5	0	0	0	0	3	Yes
83L51FC	32K ROM	256	12, 16, 20*	32	3	1	7	5	0	0	0	0	1	Yes
87L51FC	32K OTP ROM	256	12, 16, 20*	32	3	1	7	5	0	0	0	0	3	Yes
8XC51GX Product Line														
80C51GB	ROMLESS	256	12, 16	48	3	1	15	10	8	1	0	0	-	Yes
83C51GB	8K ROM	256	12, 16	48	3	1	15	10	8	1	0	0	1	Yes
87C51GB	8K EPROM	256	12, 16	48	3	1	15	10	8	1	0	0	3	Yes
8XC152 Product Line*														
80C152JA	ROMLESS	256	16.5	40	2	1	11	0	0	1	1	2	-	Yes
80C152JB	ROMLESS	256	16.5	56	2	1	11	0	0	1	1	2	-	Yes
83C152JA	8K ROM	256	16.5	40	2	1	11	0	0	1	1	2	0	Yes
8XC51SL Product Line*														
80C51SL-BQ	ROMLESS	256	16	24	2	1	10	0	4	0	1	0	-	Yes
81C51SL-BQ	8K ROM	256	16	24	2	1	10	0	4	0	1	0	0	Yes
83C51SL-BQ	8K ROM	256	16	24	2	1	10	0	4	0	1	0	0	Yes
80C51SL-AH	ROMLESS	256	16	24	2	1	10	0	4	0	1	0	-	Yes
81C51SL-AH	16K ROM	256	16	24	2	1	10	0	4	0	1	0	0	Yes
83C51SL-AH	16K ROM	256	16	24	2	1	10	0	4	0	1	0	0	Yes
87C51SL-AH	16K EPROM	256	16	24	2	1	10	0	4	0	1	0	0	Yes
80C51SL-LAL	ROMLESS	256	16	24	2	1	10	0	4	0	1	0	-	Yes
81C51SL-LAL	16K ROM	256	16	24	2	1	10	0	4	0	1	0	0	Yes
83C51SL-LAL	16K ROM	256	16	24	2	1	10	0	4	0	1	0	0	Yes
87C51SL-LAL	16K EPROM	256	16	24	2	1	10	0	4	0	1	0	0	Yes

ROM/OTP ROM/EPROM (bytes):

Speed (MHz):

Lock Bits:

8XC152 Product Line*

8XC51SL Product Line*

*ROM = System/Soft Standard BIOS

24 = 24 MHz internal-only operation

20* = 20MHz Available for Commercial Temperature Range Only

1* = 1 Lock Bit for 20MHz & 24MHz parts, no Lock Bit for 12 & 16MHz parts

P = Program verification disabled, external memory access limited to 4K

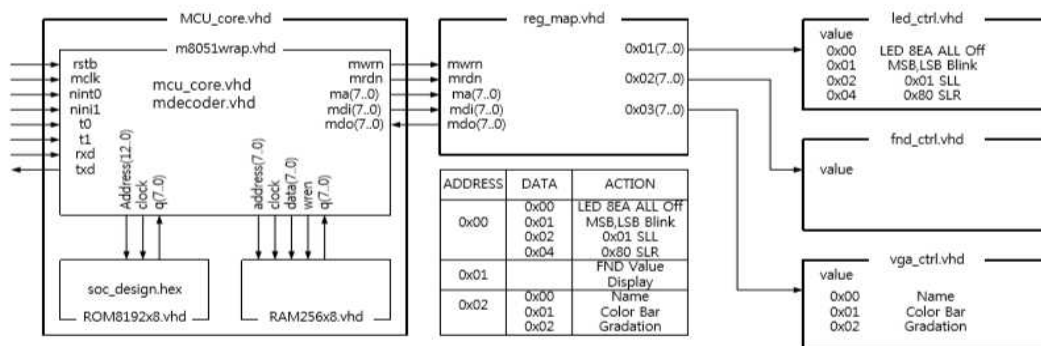
= Communication Controller

= Keyboard Controller

2)8051 실습 순서

실습하고자 하는 프로그램간 연결도를 확인할 수 있다. 메인core를 reg_map과 연결되어 각 모듈과 연결하는 것이 목표였으나 시간관계상 이번 실습에서는 mcu코어의 제어까지만 하는 것을 목표로 한다.

실습 구조



실습 순서

- 8051 VHDL core 압축해제 및 디렉토리 설정
- 프로젝트 생성
- ROM 및 RAM 생성
- 계층구조에 따른 파일 구조 수정
- 8051 CPU Core & F/W 동작 확인
- 레지스터 맵 및 Special Function Logic 작성
- 8051 core와 작성한 레지스터 맵, SFL 연동 확인

즉 8051Core&F/W 동작확인 까지를 목표로 한다.

2. 소스코드 설명

각 실습 단계들을 통해 프로젝트를 작성하고 진행한다.

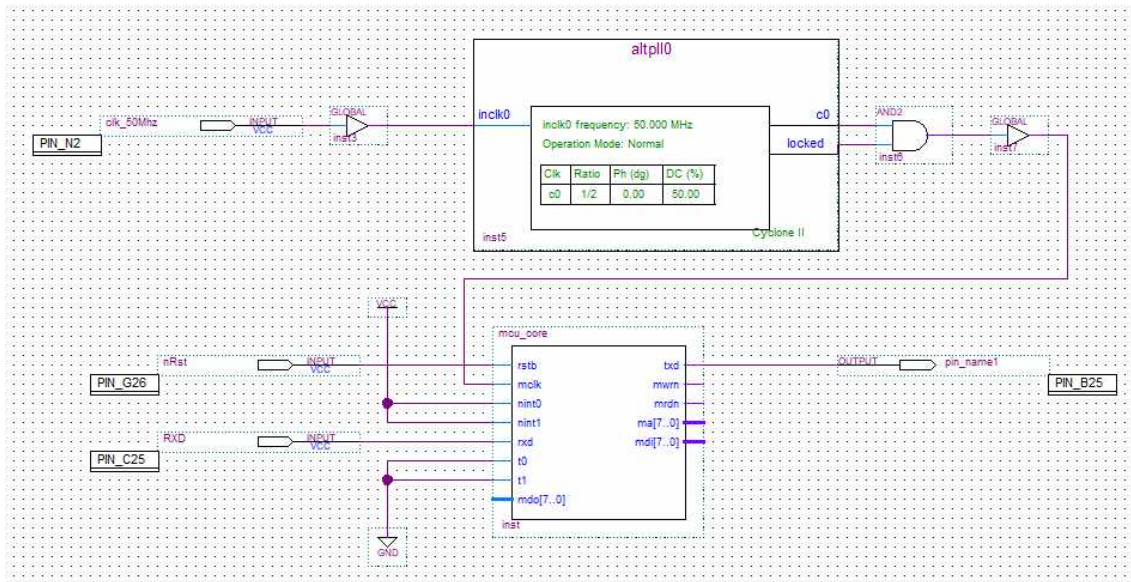
1. 프로젝트 생성. 8051코어파일들과 Wrap파일 추가.
2. Ram 생성
3. Rom 생성
4. MCU_Core_component 추가
5. Top_Design 설계

1)MCU_Core_component

```
104 component ROM8192x8
105 port (
106     address : in std_logic_vector(12 downto 0);
107     clock : in std_logic;
108     q : out std_logic_vector( 7 downto 0)
109 );
110 end component;
111
112 component RAM256x8
113 port (
114     address : in std_logic_vector( 7 downto 0);
115     clock : in std_logic;
116     data : in std_logic_vector( 7 downto 0);
117     wren : in std_logic;
118     q : out std_logic_vector( 7 downto 0)
119 );
120 end component;
121
219 not_idm_nfwe <= not idm_nfwe;
220
221 prg_ram : ROM8192x8
222 port map (
223     address => ipm_a(12 downto 0),
224     clock => mclk,
225     q => ipm_do
226 );
227
228 sfr_ram : RAM256x8
229 port map (
230     address => idm_a,
231     clock => mclk,
232     data => idm_di,
233     wren => not_idm_nfwe,
234     q => idm_do
235 );
236
237 end rtl;
```

MCU코어에 ROM과 RAM을 추가해준다.

2)8051 Core& F/W Top Design



단순하게 MCU코어에 CLK과 PC와의 통신, 리셋버튼만을 설정하여 동작을 확인하게 한다.

3. 시뮬레이션 결과 및 설명

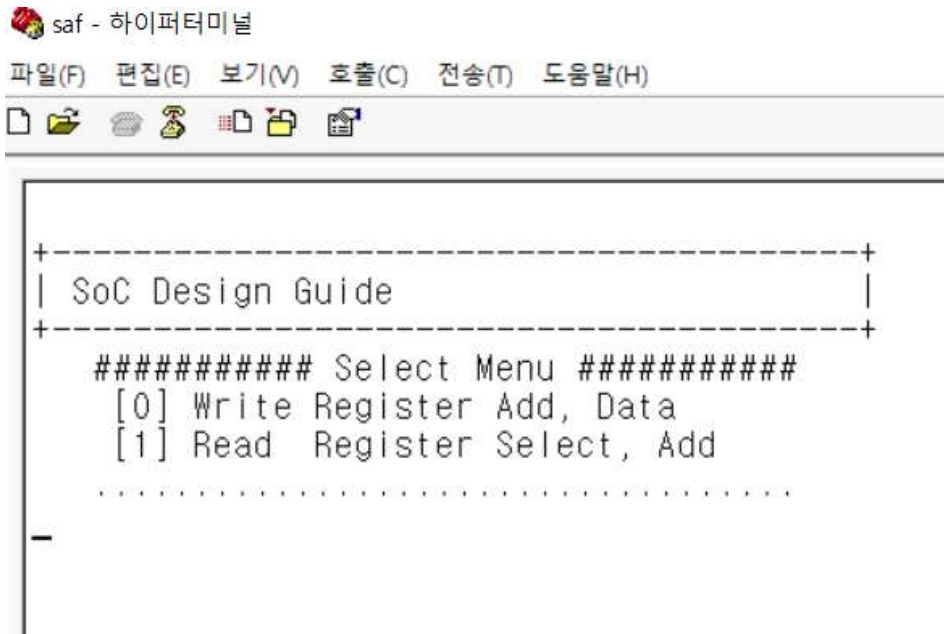
-8051 Core& F/W 동작확인

단순하게 8051CORE의 동작을 확인하려 하기 때문에 별도의 레지스터 데이터가 있지 않으므로 코어에 설정된 동작인 WRITE or READ동작을 인식하는 것 까지 확인하고자 한다.

4. DE2 실습

-8051 Core& F/W 동작확인

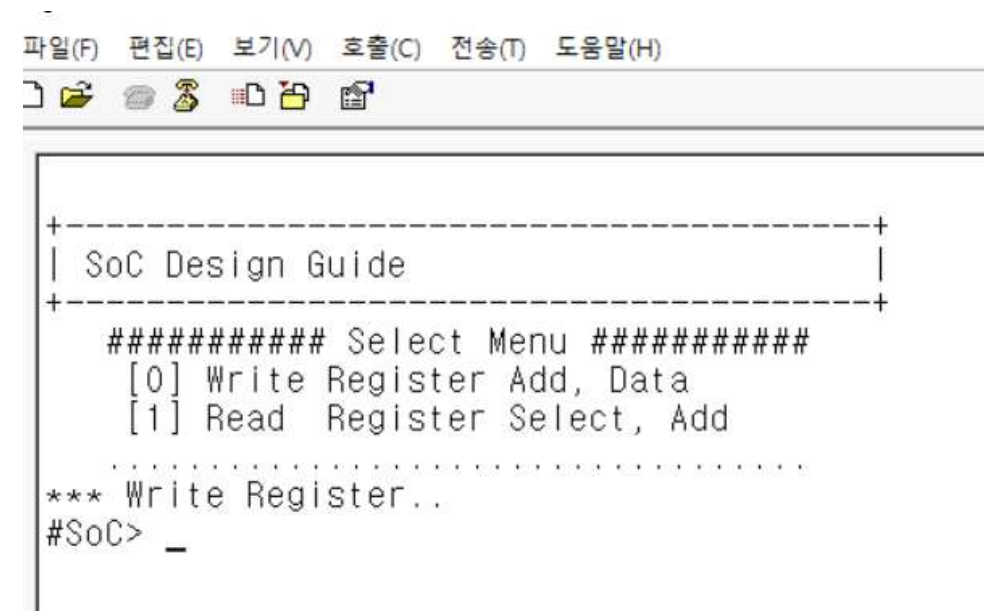
1. DE2보드와 PC를 연결하였을 때 확인할 수 있는 화면이다.
읽기(1) 혹은 쓰기(0) 동작을 선택할 수 있다.



The screenshot shows a HyperTerminal window titled 'saf - 하이퍼터미널'. The menu bar includes '파일(F)', '편집(E)', '보기(V)', '호출(C)', '전송(T)', and '도움말(H)'. The main display area shows a menu with dashed borders. The text inside the menu is as follows:

```
+-----+
| SoC Design Guide                               |
+-----+
##### Select Menu #####
[0] Write Register Add, Data
[1] Read  Register Select, Add
.....
-
```

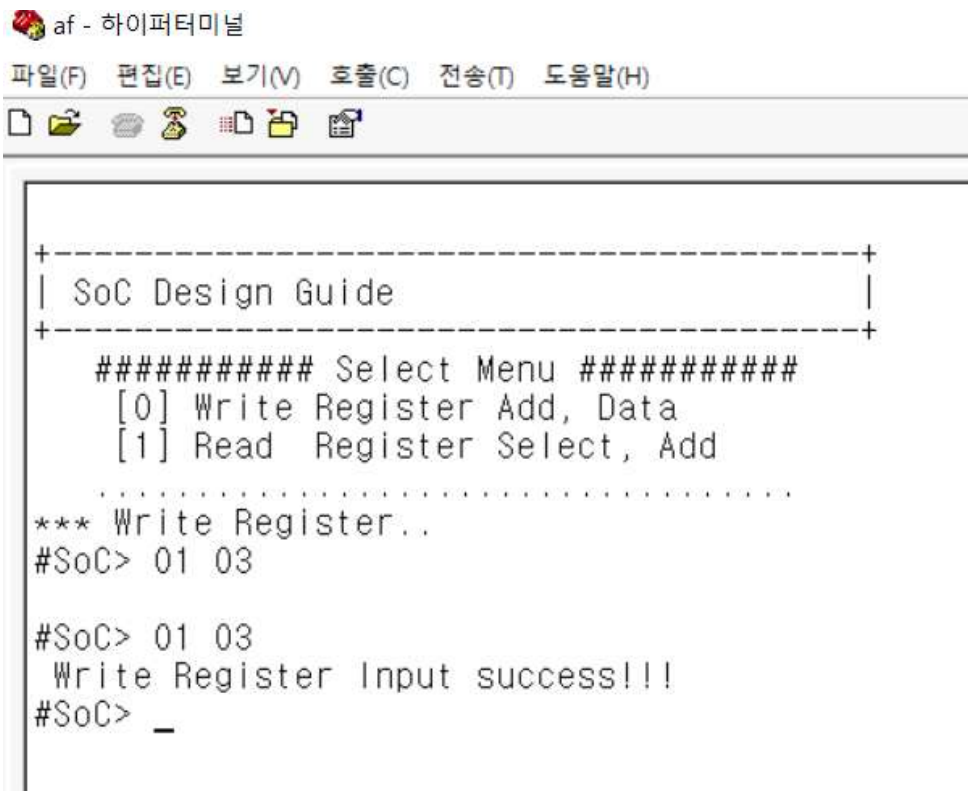
2. 쓰기(0)를 눌렀을 때 나오는 화면이다. PC에서 0 입력시 나타나는 화면이다. 쓰기상태 동작을 한다는 메시지가 출력된다



The screenshot shows the same HyperTerminal window as before, but with additional output text. The menu is still visible, and the following text has been added to the display:

```
*** Write Register..
#SoC> -
```


3. 쓰기동작시 01번 레지스터에 03데이터를 쓰는 동작을 하였다.
해당 메시지 입력시 쓰기에 성공했다고 메시지 출력이 나타난다. 하지만
진짜 입력은 되지 않는다. 이유는 코어의 동작만 확인할 뿐
레지스터까지 연결을 해놓지 않았기 때문이다.



```
af - 하이퍼터미널
파일(F) 편집(E) 보기(V) 호출(C) 전송(T) 도움말(H)
[Icons]

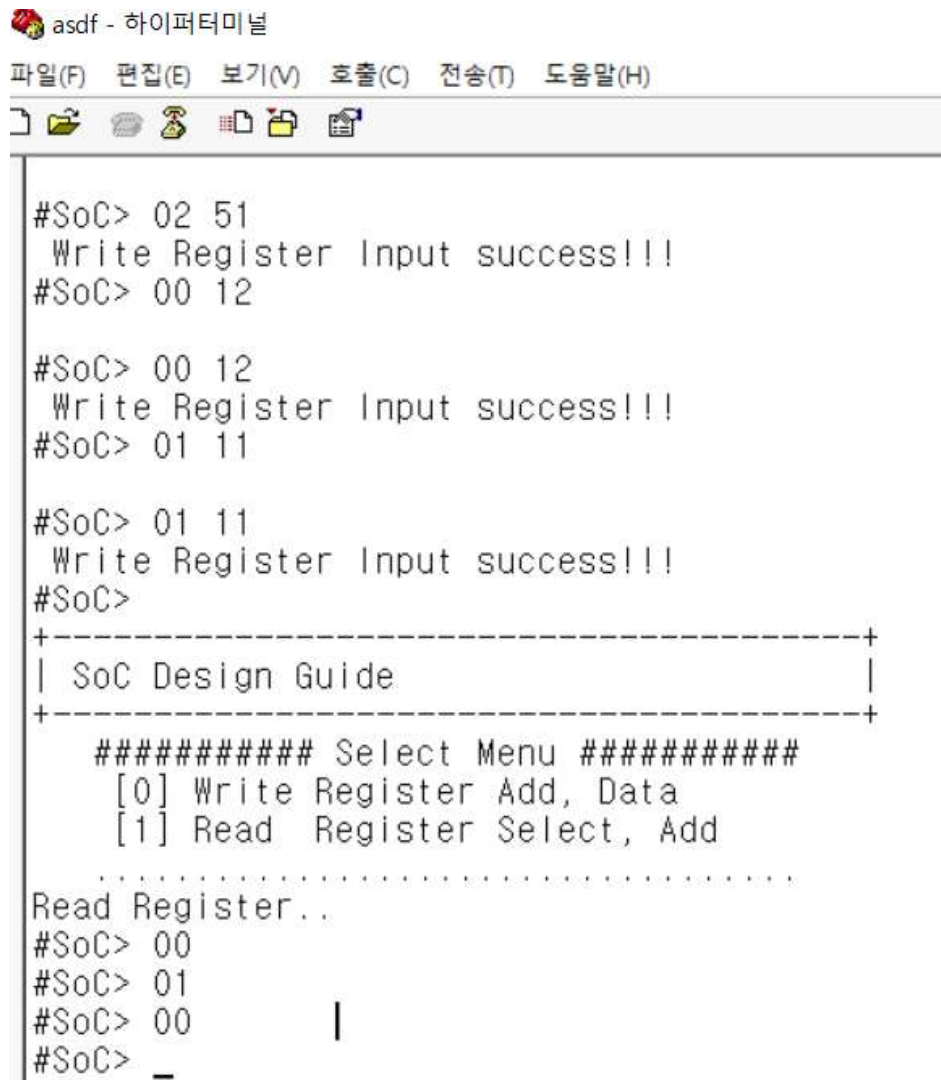
+-----+
| SoC Design Guide |
+-----+
##### Select Menu #####
[0] Write Register Add, Data
[1] Read Register Select, Add
.....
*** Write Register..
#SoC> 01 03

#SoC> 01 03
Write Register Input success!!!
#SoC> _
```

4. 처음 화면에서 읽기(1)를 눌렀을 때 나오는 화면이다. PC에서 1 입력시 나타나는 화면이다. 읽기상태 동작을 한다는 메시지가 출력된다.

```
asdf - 하이퍼터미널
파일(F) 편집(E) 보기(V) 호출(C) 전송(T) 도움말(H)
[Icons]
#SoC> 01 11
Write Register Input success!!!
#SoC>
+-----+
| SoC Design Guide |
+-----+
##### Select Menu #####
[0] Write Register Add, Data
[1] Read Register Select, Add
.....
Read Register..
#SoC> 00
#SoC> 01
#SoC> 00
#SoC>
+-----+
| SoC Design Guide |
+-----+
##### Select Menu #####
[0] Write Register Add, Data
[1] Read Register Select, Add
.....
Read Register..
#SoC> _
```

5. 읽고자 하는 레지스터 번호를 입력해 보지만 읽혀지지 않는 화면이다. 이는 읽기 동작만 실행될 뿐 레지스터맵을 연결해 놓지 않았기 때문에 읽히는 데이터가 없어서 그렇다.



```
asdf - 하이퍼터미널
파일(F)  편집(E)  보기(V)  호출(C)  전송(T)  도움말(H)

#SoC> 02 51
Write Register Input success!!!
#SoC> 00 12

#SoC> 00 12
Write Register Input success!!!
#SoC> 01 11

#SoC> 01 11
Write Register Input success!!!
#SoC>
+-----+
| SoC Design Guide |
+-----+
##### Select Menu #####
[0] Write Register Add, Data
[1] Read Register Select, Add
.....
Read Register..
#SoC> 00
#SoC> 01
#SoC> 00      |
#SoC> _
```

6. DE2보드의 SW0입력시 리셋이 되는 화면이다. 초기 화면으로 리셋동작 후 PC로 리셋 완료 메시지를 출력한다.

```

asdf - 하이퍼터미널
파일(F) 편집(E) 보기(V) 호출(C) 전송(T) 도움말(H)

#SoC> 01 11
Write Register Input success!!!
#SoC>
+-----+
| SoC Design Guide |
+-----+
##### Select Menu #####
[0] Write Register Add, Data
[1] Read Register Select, Add
.....
Read Register..
#SoC> 00
#SoC> 01
#SoC> 00
#SoC>
+-----+
| SoC Design Guide |
+-----+
##### Select Menu #####
[0] Write Register Add, Data
[1] Read Register Select, Add
.....

```

5. 토의 및 실습소감

8051코어가 기본적인 SoC를 실현한 칩이라고 소개해 주셔서 흥미있게 실습을 진행하였습니다. 비록 우리가 만든 레지스터맵을 붙여서 실험을 해보지는 않았으나 ROM과 RAM 설정을 통해 기본적인 코어 동작을 하는 것이 신기하였습니다. 기말고사가 다가옴으로써 다른 실습을 하지 않은 점 아쉬웠습니다. 한 학기동안 많은 것을 배워가는 느낌입니다. 감사합니다.