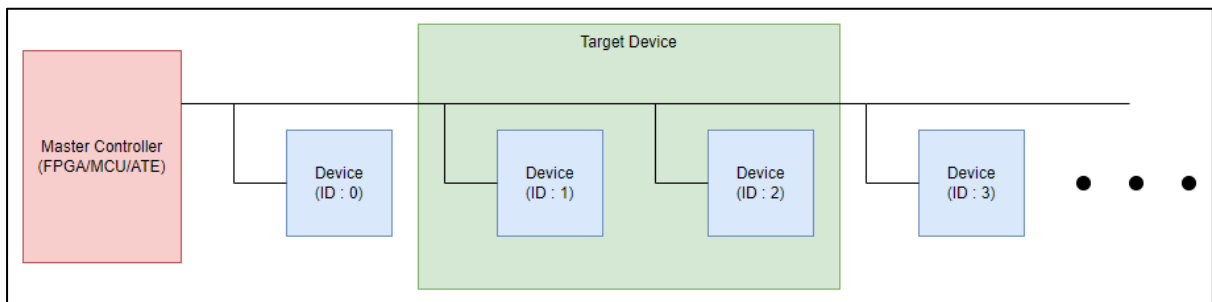


Emulator 사용법

Emulator 프로그램은 (주)LeoLSI에서 생산된 모든 제품들에 대한 Logic적인 사용법을 보여주기 위해서 제작된 프로그램이다.

각 Device의 물리적인 연결에 대해서는 향후 회로도를 보여줄 수 있도록 하자.

기본 회로 구성은 아래와 같다.(제품별로 회로 구성이 다르므로, 향후 수정이 필요함.)



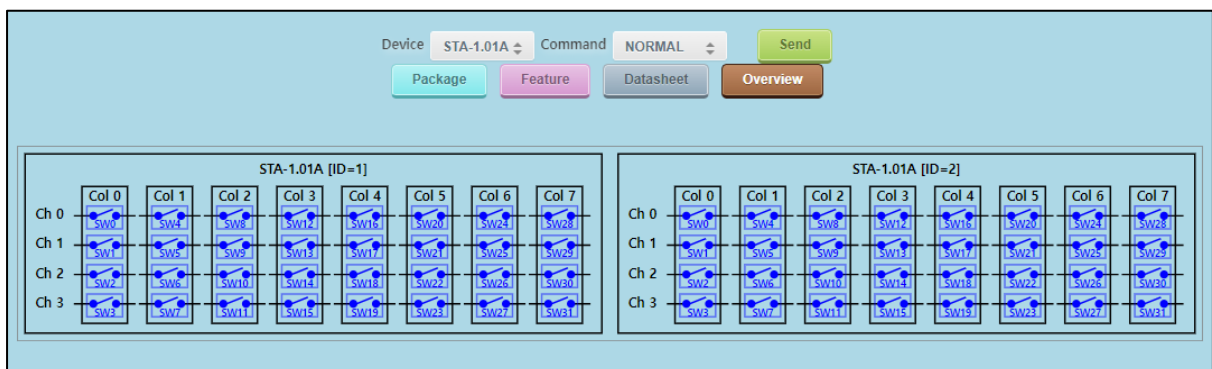
Emulator 프로그램은 "Master Controller"와 "Target Device"에 대한 동작들을 설명한다. **향후 Target Device의 개수를 늘릴 수 있도록 수정이 필요함.**

Emulator 실행.

Emulator를 실행하기 위해서는 "emulator"폴더에 있는 index.html 파일을 더블 클릭하거나 브라우저(크롬, Microsoft Edge, 사파리, 등)에서 Open하여 실행한다. HTML5 규격으로 작성되었기 때문에 대부분의 브라우저에서 실행된다. 따라서, 사용자는 브라우저가 설치된 장치라면, OS(Windows, Linux, Android, Mac OSX)에 상관없이 모든 장치에서 사용할 수 있다.

회사 Web Page에 Link를 만든다면, 인터넷이 되는 곳에서는 언제든지 연결(사용)할 수 있다.

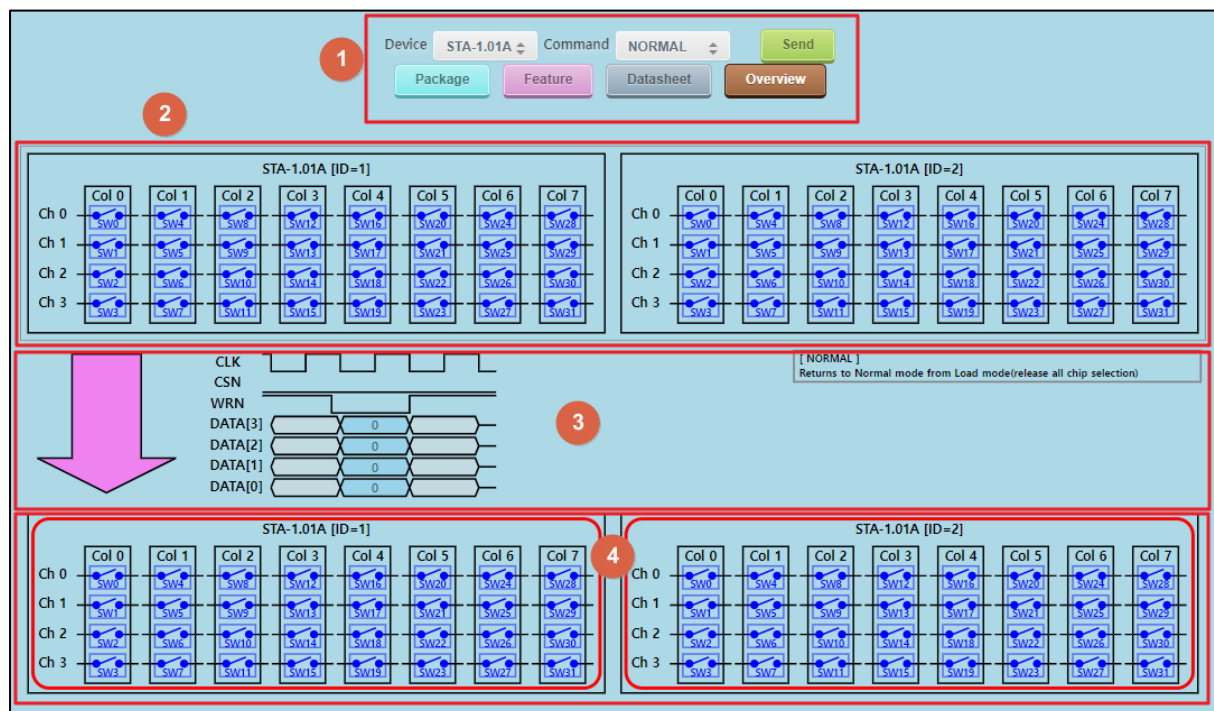
실행(접속) 결과는 아래와 같은 화면이 출력된다.(페이지가 열린다.)



기본 UI 설명

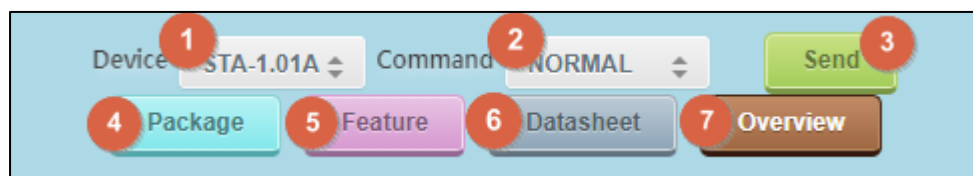
기본 화면 구성은 아래 그림과 같이 4개의 그룹으로 나눌 수 있다.

1. UI(User Interface) 화면.
2. Pre Status(명령 실행 이전 상태) 화면
3. 명령 설명 화면
4. Next Status(명령 실행 이후 상태) 화면



UI 화면

Emulator 프로그램의 다양한 동작 실행하거나 선택하는 화면이다. 메인 컨트롤 화면이라고 볼 수 있다.



각 UI(버튼)의 기능에 대해서 알아보자.

1. Device 선택 버튼
프로그램에서 지원하는 모든 Device의 목록 중에서 Emulation할 Device를 선택한다.
Device 목록은 그룹으로 구성되어 있으며, 제품의 특성 별로 그룹화되어 있다.

2. Command 선택 버튼

선택된 Device에 대한 모든 Command 목록이 있으며, 각 명령을 선택할 수 있다.

선택된 명령에 따라서 추가적인 입력을 요구하는 UI가 출력된다. 추가적인 입력에 대해서는 Device의 Datasheet를 참조해야 한다.

3. Send 실행 버튼

실행할 Command가 모두 설정되었다면, "Send" 버튼을 클릭하여 Command가 실행할 수 있다.

4. Package 버튼

"Device 선택"에서 선택된 Device의 Package 정보를 보여준다.

5. Feature 버튼

"Device 선택"에서 선택된 Device의 Feature 정보를 보여준다.

6. Datasheet 버튼

"Device 선택"에서 선택된 Device의 Datasheet를 보여준다.

7. Overview 버튼

"Device 선택"에서 선택된 Device와 같은 그룹에 있는 다른 Device들과의 Overview 정보를 보여준다.

Pre status 화면

Pre status 화면은 선택한 명령(Command)가 실행되지 전의 상태를 보여준다.

사용자의 명령이 실행될 때 마다 Device의 이전 상태가 Update된다.

명령 실행 화면

사용자가 선택한 명령(Command)에 대한 간략한 동작들을 보여준다. UI 화면에서 "Send" 버튼을 클릭해야만 한다.

좌측에는 Interface signal에 대한 timing을 보여준다. 즉, Master Controller가 Device로 입력되는 Timing을 보여준다.

우측에는 선택된 명령(Command)에 대한 설명이 출력된다.

Next status 화면

UI 화면에서 "Send" 버튼을 클릭해야만 한다.

사용자가 선택한 명령(Command)을 실행한 결과를 보여준다.

Pre status 화면에서 사용자가 선택한 명령이 실행된 결과를 보여 주기 때문에 사용자는 각 명령이 어떻게 동작되는지 바로 바로 확인할 수 있다.

1세대 STA 제품군.

1세대 STA 제품군에는 "STA-1.01A", "STA-2", "STA-2.01", "STA-3B" 라는 4개의 제품이 있다.

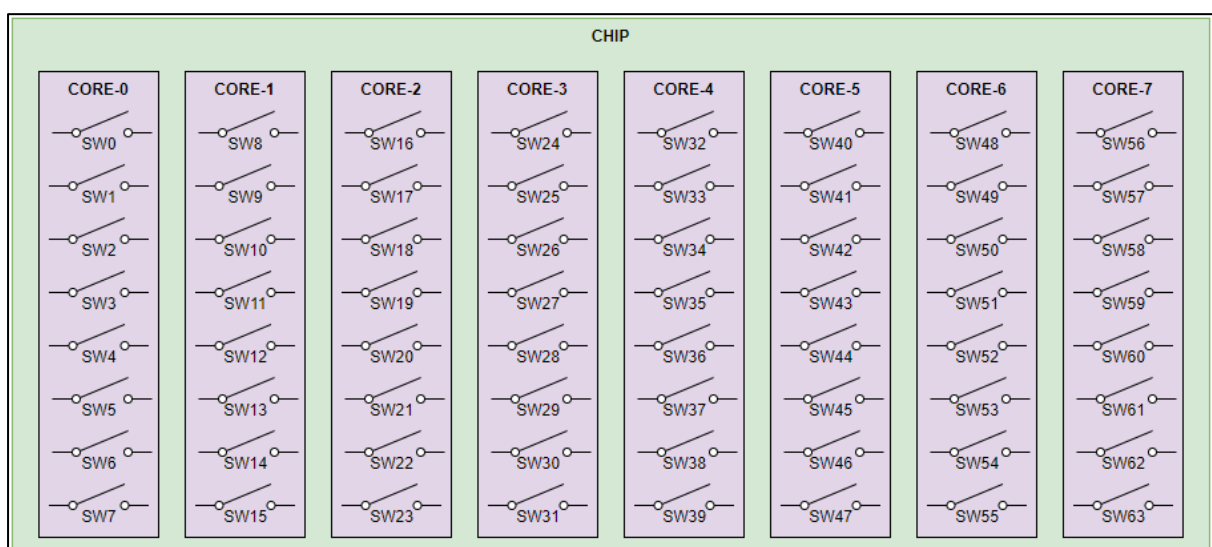
모든 제품의 Logic Interface와 Command는 동일하다, 하지만, Device별로 Switch Channel의 구성이 다르기 때문에 약간의 차이점을 확인할 수 있다.

1세대 STA Device들의 ID는 CHIP 내의 OTP 영역에 Program되어있기 때문에 Device를 구매할 때 필요한 ID를 갖는 Device를 구매해야 한다.

Command List 설명.

1세대 STA 제품들은 모두 동일한 Command List를 사용한다.

모든 Switch는 8개의 Core로 구성되고, 각 Code는 4 / 8개의 Switch들의 묶음으로 구성된다.



Command는 2가지 유형으로 구분된다. "1-Clock" Command와 "2-Clock" Command이다.

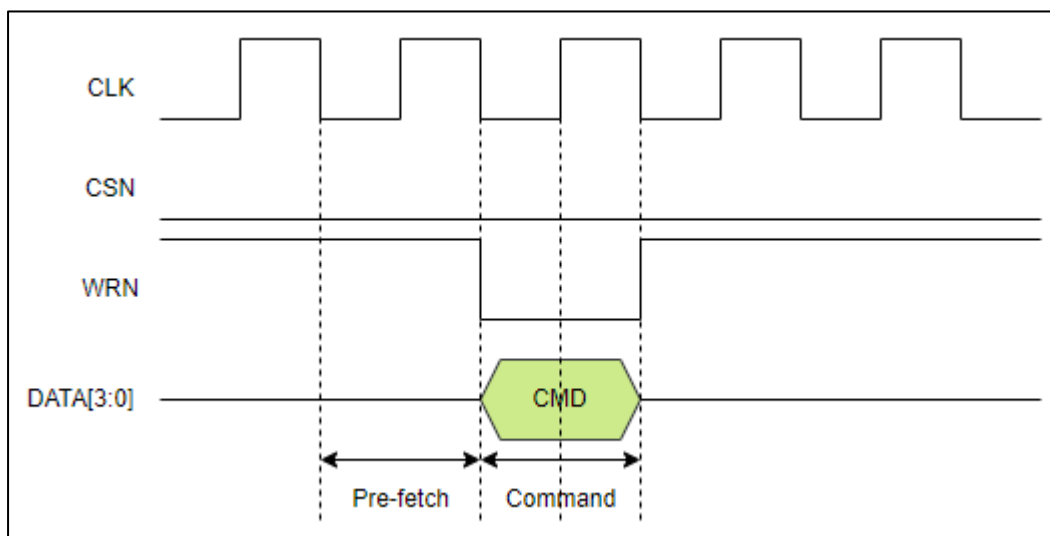
각 유형은 WRN 신호가 Active-Low일 때 CLK 클럭의 개수로 구별된다. 즉, "1-Clock" Command는 CLK 클럭이 1개이고, "2-Clock" Command는 CLK 클럭이 2개 발생한다.

자세한 Command Timing은 Datasheet를 참조하자.

각각의 명령(Command)에 대해 알아보자.

"1-Clock" Command

아래와 같은 Timing으로 Command가 입력된다.



"1-Clock" Command에는 6개의 명령이 있다. 3개 명령은 Chip Test를 위한 명령이며, 나머지 3개 명령만 사용해야 한다.

1. NORMAL(0x0)
2. LOAD_ALL(0x1)
3. VIRTUAL(0x2)
4. CLEAR_ALL(0x3)

CLEAR_ALL(0x3) 명령은 동일한 Interface bus(STA Interface)에 연결된 모든 Chip에 대해서 Reject되지 않은 모든 Switch를 OFF시킨다. Reject된 Switch는 Control되지 않기 때문에 Reject상태를 그대로 유지하고, Switch는 OFF 상태를 유지한다.

5. ENABLE_ALL(0x4)

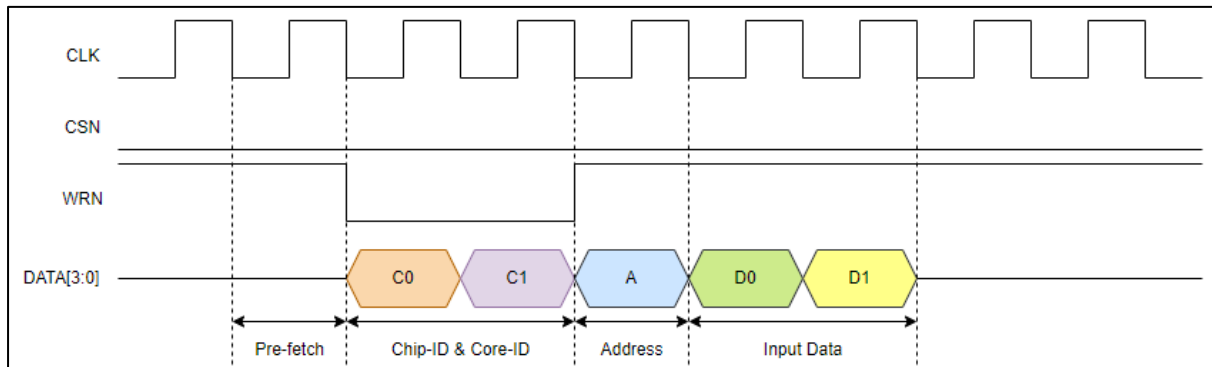
이 명령은 동일한 Interface bus(STA Interface)에 연결된 모든 Chip에 대해서 Reject되지 않은 모든 Switch를 ON시킨다. Reject된 Switch는 Control되지 않기 때문에 Reject상태를 그대로 유지하고, Switch는 OFF 상태를 유지한다.

6. INITIAL_ALL(0x5)

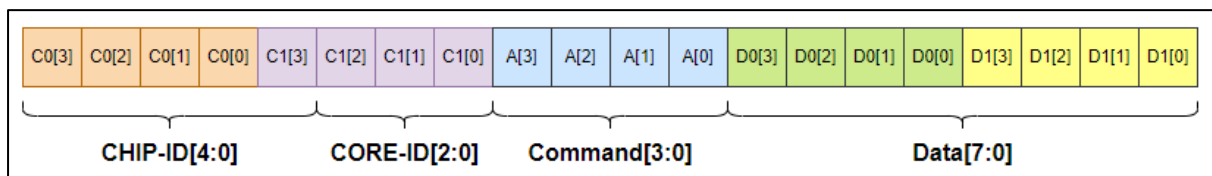
이 명령은 동일한 Interface bus(STA Interface)에 연결된 모든 Chip에 대해서 모든 Switch를 ON시킨다. Reject된 Switch는 Reject상태를 Release하고, ON된다.

“2-Clock” Command

아래와 같은 Timing으로 Data가 입력된다.



5-cycle에 입력된 Data는 아래와 같이 Command Frame으로 처리된다.



“2-clock” Command의 Name에 따라 3가지 유형을 갖는다. 명령어 이름에 따라서 아래와 같은 동작을 한다.

- CHL(Chip-Level): Command Frame에서 Chip-ID와 Core-ID는 무시되고, 모든 Chip의 모든 Core에 대해서 명령이 실행된다.
- CRL(Core-Level): Command Frame에서 Core-ID는 무시되고, Chip-ID와 같은 ID를 갖는 Chip에 대해서 모든 Core에 명령이 실행된다.
- SWL(Switch-Level): Command Frame에서 Chip-ID와 같은 ID를 갖는 Chip에 대해서 Core-ID와 같은 Core에만 명령이 실행된다.

자세한 Command Frame은 Datasheet를 참조하자.

1. AND_CRL(0x0)

이 명령은 “CRL” 유형의 명령으로 특정 chip에 대해서 모든 Core에 명령이 Data[7:0]를 실행된다.

이 명령의 동작은 선택된 Chip에 모든 Core에 대해서 각 현재 Switch의 상태와 Data[7:0]을 bitwise AND를 실행한 결과를 Switch에 적용한다. 즉, 특정 Chip에 대해서 각 Core단

위로 Data[7:0]의 bit 값이 "0"인 Switch를 OFF 시킨다.

예)

2. OR_CRL(0x1)

이 명령은 "CRL" 유형의 명령으로 특정 chip에 대해서 모든 Core에 명령이 Data[7:0]를 실행된다.

이 명령의 동작은 선택된 Chip에 모든 Core에 대해서 각 현재 Switch의 상태와 Data[7:0]을 bitwise OR를 실행한 결과를 Switch에 적용한다. 즉, 특정 Chip에 대해서 각 Core단위로 Data[7:0]의 bit 값이 "1"인 Switch를 ON 시킨다.

예)

3. DIRECT_CRL(0x2)

이 명령은 "CRL" 유형의 명령으로 특정 chip에 대해서 모든 Core에 명령이 Data[7:0]를 실행된다.

이 명령의 동작은 선택된 Chip에 모든 Core에 대해서 Data[7:0] 값을 Switch에 적용한다. 즉, 특정 Chip에 대해서 각 Core단위로 Data[7:0]의 bit 값이 "1"이면 Switch를 ON 시키고, "0"이면 OFF 시킨다.

예)

4. DIRECT_CHL(0x3)

이 명령은 "CHL" 유형의 명령으로 모든 Chip에 대해서 모든 Core에 명령이 Data[7:0]를 실행된다.

이 명령의 동작은 모든 Chip에 모든 Core에 대해서 Data[7:0] 값을 Switch에 적용한다. 즉, 모든 Chip에 대해서 각 Core단위로 Data[7:0]의 bit 값이 "1"이면 Switch를 ON 시키고, "0"이면 OFF 시킨다.

예)

5. REJECT_CRL(0x4)

이 명령은 "CRL" 유형의 명령으로 특정 chip에 대해서 모든 Core에 명령이 Data[7:0]를 실행된다.

이 명령의 동작은 선택된 Chip에 모든 Core에 대해서 현재 REJECT 상태와 Data[7:0]을 bitwise AND 결과를 Switch에 적용한다.(bit 값이 "0"이면 REJECT이다.) 즉, 특정 Chip에 대해서 각 Core단위로 Data[7:0]의 bit 값이 "0"이면 해당 Switch를 REJECT(OFF) 시키고, "1"이면 이전 상태를 유지한다.

예)

6. AND_SWL(0x8)

이 명령은 "SWL" 유형의 명령으로 선택된 chip에서 선택된 Core에 Data[7:0]를 실행된다.

이 명령의 동작은 선택된 Chip에서 선택된 Core에 대해서 현재 각 Switch의 상태와 Data[7:0]을 bitwise AND를 실행한 결과를 Switch에 적용한다. 즉, 선택된 Chip의 선택된 Core에서 Data[7:0]에서 bit 값이 "0"인 Switch들을 OFF시킨다.

예)

7. OR_SWL(0x9)

이 명령은 "SWL" 유형의 명령으로 선택된 chip에서 선택된 Core에 Data[7:0]를 실행된다.

이 명령의 동작은 선택된 Chip에서 선택된 Core에 대해서 현재 각 Switch의 상태와 Data[7:0]을 bitwise OR를 실행한 결과를 Switch에 적용한다. 즉, 선택된 Chip의 선택된 Core에서 Data[7:0]에서 bit 값이 "1"인 Switch들을 ON시킨다.

예)

8. DIRECT_SWL(0xA)

이 명령은 "SWL" 유형의 명령으로 선택된 chip에서 선택된 Core에 Data[7:0]를 실행된다.

이 명령의 동작은 선택된 Chip에서 선택된 Core에 대해서 Data[7:0]을 Switch에 적용한다. 즉, 선택된 Chip의 선택된 Core에서 Data[7:0]에서 bit 값이 "1"이면 Switch를 ON시키고, "0"이면 OFF시킨다.

예)

9. REJECT_SWL(0xC)

이 명령은 "SWL" 유형의 명령으로 선택된 chip에서 선택된 Core에 Data[7:0]를 실행된다.

이 명령의 동작은 선택된 Chip에서 선택된 Core에 대해서 현재 각 Switch의 REJECT 상태와 Data[7:0]을 bitwise AND를 실행한 결과를 Switch Reject에 적용한다. 즉, 선택된 Chip의 선택된 Core에서 Data[7:0]에서 bit 값이 "0"이면 Switch를 REJECT(OFF)시킨다.

예)

10. CANCEL_RJT(0xD)

이 명령은 모든 Chip에 대해서 선택된 Core의 모든 Switch Reject상태를 Cancel(Release)한다. -> 모든 Chip인지 선택된 Chip인지 확인되지 않았다.

이 명령의 동작은 모든 Chip에서 선택된 Core에 대해서 모든 Switch의 REJECT 상태를 Cancel(Release)한다. 단, Reject상태가 Cancel되었지만, Switch는 OFF되어 있다. Reject이 Cancel된 Switch는 ON/OFF Control이 가능해진다.

예)

STA-1.01A

STA-2

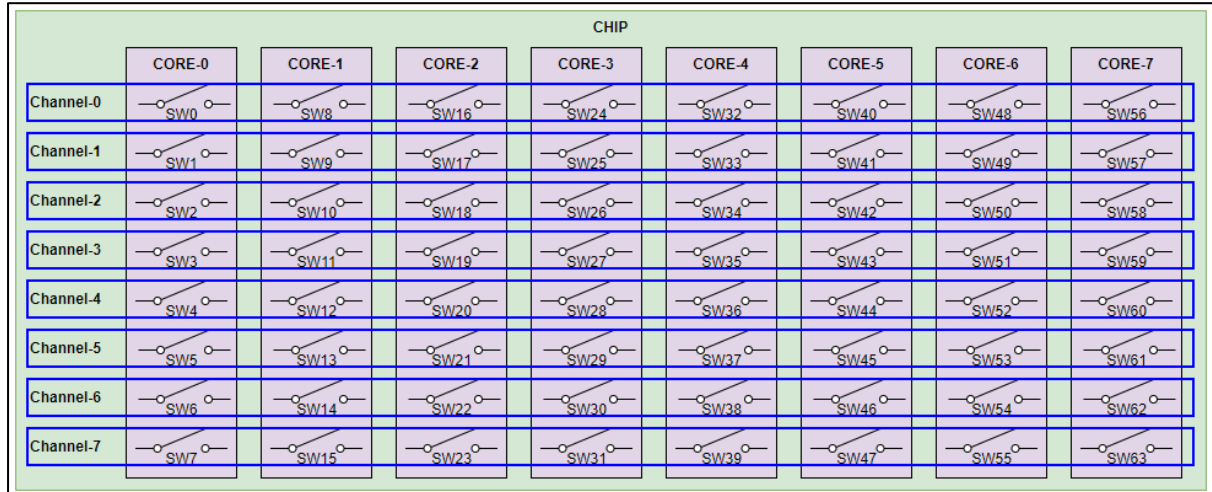
STA-2.01

STA-3B

2세대 STA 제품군

2세대 STA 제품들은 1세대 제품과 다른 Command Frame을 사용한다.

각 Device의 Switch들은 Core[7:0]와 Channel[7:0]에 의해서 Matrix 구조로 구성되어있다.



STA Interface에서 Command는 2가지 유형으로 구분된다.

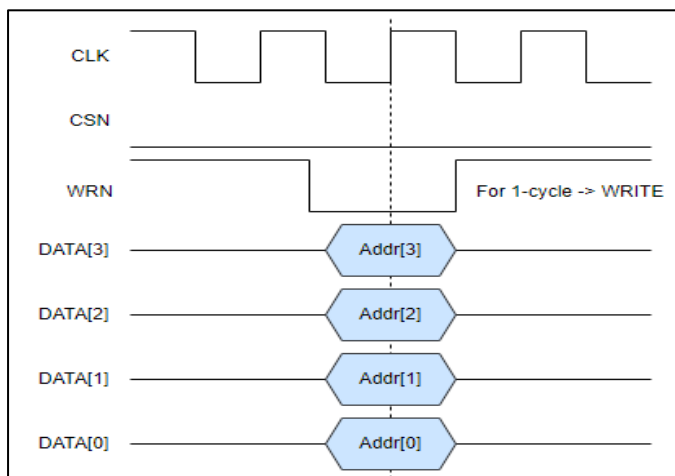
- "1-Clock" Command: WRN의 Active Low 구간이 1-cycle인 명령.
- "2-Clock" Command: WRN의 Active Low 구간이 2-Cycles인 명령.

자세한 Command Timing은 Datasheet를 참조하자.

각 Command 유형에 따라서 어떤 Command가 있는지 확인해 보자.

"1-Clock" Command

"1-clock" Command Timing은 아래와 같다. Addr[3:0] 값에 따라서 4가지 명령이 실행된다.



1. RESET_ALL(0x2)

동일 Interface에 연결된 모든 Chip의 모든 Switch의 Reject를 Cancel하고, OFF시킨다.

2. CLEAR_ALL(0x3)

동일 Interface에 연결된 모든 Chip의 Reject되지 않은 모든 Switch를 OFF시킨다. Reject된 Switch는 Reject 상태를 유지한다.

3. ENABLE_ALL(0x4)

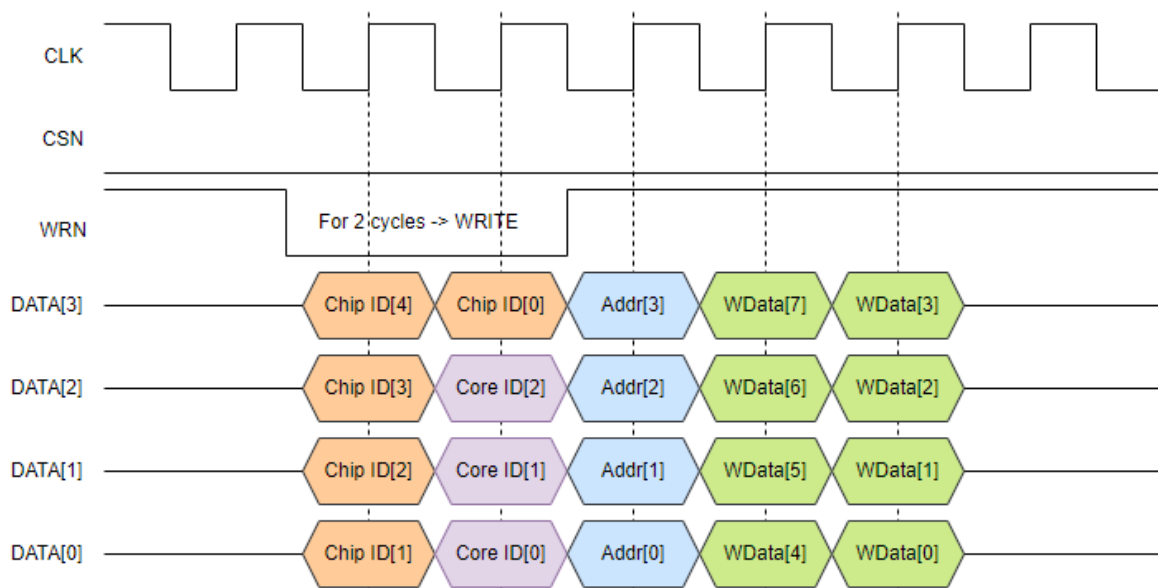
동일 Interface에 연결된 모든 Chip의 Reject되지 않은 모든 Switch를 ON시킨다. Reject된 Switch는 Reject 상태를 유지하며 OFF된다.

4. INITIAL_ALL(0x5)

동일 Interface에 연결된 모든 Chip의 모든 Switch의 Reject를 Cancel하고, ON시킨다.

"2-Clock" Command

"2-clock" Command Timing은 아래와 같다. Addr[3:0] 값에 따라서 10가지 명령이 실행된다.



"2-clock" Command는 Command Name에 따라서 4가지 유형으로 분류된다.

- *_CHP_COR

Command Frame에서 Chip ID는 사용되고, Core ID는 무시된다.

Command Frame에 기술된 Chip ID와 Device ID가 같은 Chip에 대해서 모든 Core에 명령이 실행된다.

- *_BNK_COR

Command Frame에서 Chip ID와 Core ID는 무시된다.

즉, 동일한 Interface에 연결된 모든 Chip에 대해서 모든 Core에 명령이 실행된다.

- *_CHP_CHN

Command Frame에서 Chip ID는 사용되고, Core ID는 무시된다.

Command Frame에 기술된 Chip ID와 Device ID가 같은 Chip에 대해서 모든 Channel에 명령이 실행된다.

- *_BNK_CHN

1. DIRECT_CHP_COR