ATE305 Control Guide Part 3

Daniel 2017/4/21 Initial Version

New PEB Board 進行 digital pattern 測試時我們稱為 Normal Mode，另外每一個 pin 我們都能夠設定為 PMU mode，進行 PMU 相關參數量測。

以一個基本的 ATE305 + Channel FPGA + PMU ADC + Relay 的接線如下圖:

Signal Path

Control Path

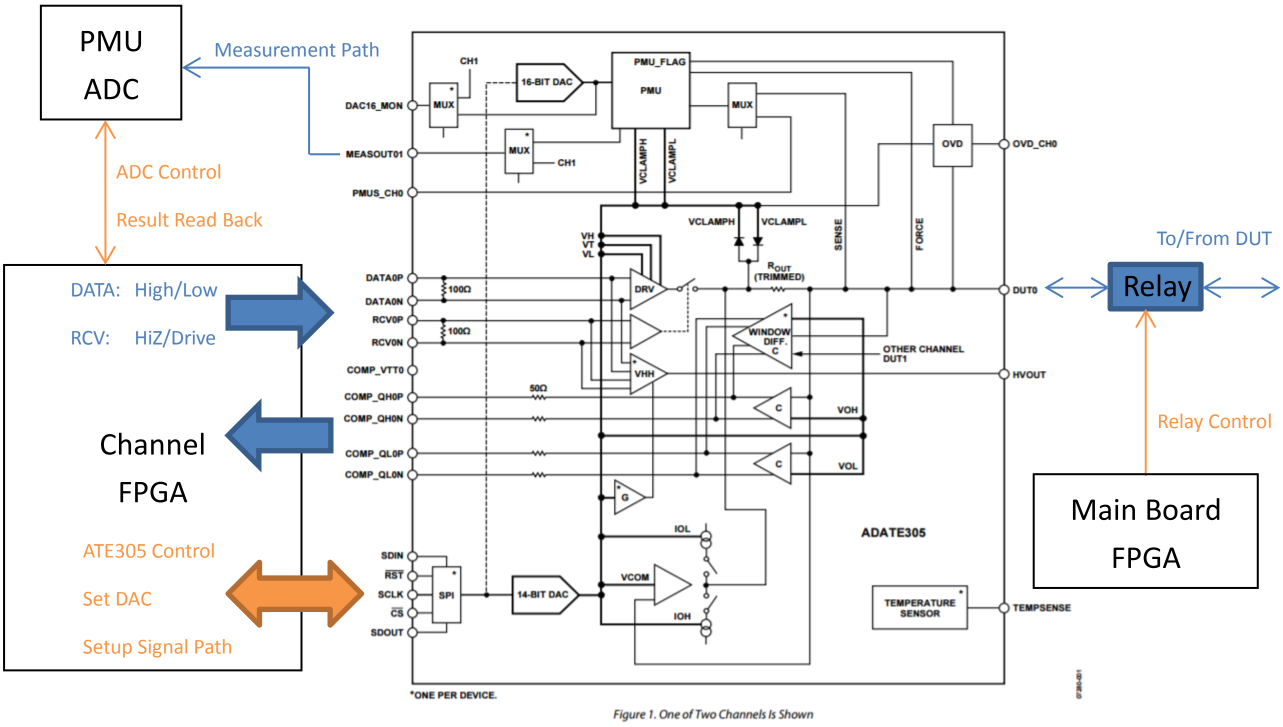


Figure 1. One of Two Channels Is Shown

Pin_Mode API:

- Pin_Mode (channel, OFF/PMU/VTT/LOAD/NORMAL)

  每個 Channel 可以個別指定 Pin Mode:

  OFF: 關閉 Pin Drive、PMU 所有功能

  PMU: 設定為 PMU 使用(此時可以 FV/FI/MV/MI)

  VTT: Terminate on VTT (此時可以 MV)

  LOAD: 加掛 Active Load (此時可以 MV)

  Normal: 關閉 Active Load (此時可以 MV)

PMU APIs:

- FV (channel, Value, I_range, clamp_H, clamp_L, delay) // default delay = 0ms
- FI (channel, Value, I_range, clamp_H, clamp_L, delay) // default delay = 0ms
- MV (channel, count) // default count(1)
- MI (channel, count) // default count(1)

Pin_Mode API 底層撰寫指導:

- Pin_Mode (channel, OFF/PMU/VTT/LOAD/NORMAL)

  Channel 對應哪個 Slot 由上層軟體決定，底層 API 我們只需決定 1~64Channel 是哪一個 Channel FPGA 的哪一個 Channel 即可:

  (channel / 16) = CH_FPGA_SEL，(channel % 16) = CH_SEL

  ```
      W    0xFF       0x0002 << CH_FPGA_SEL       // Channel FPGA Select
  ```

  基本上 Pin_Mode API 主要控制 ATE305 的 Register 0x0C、0x0D 位置

  針對 OFF Mode，我們使用 ATE305 spec Table29 的第一行 “High-Z without clamps”，並且同時關閉 PMU 的功能:

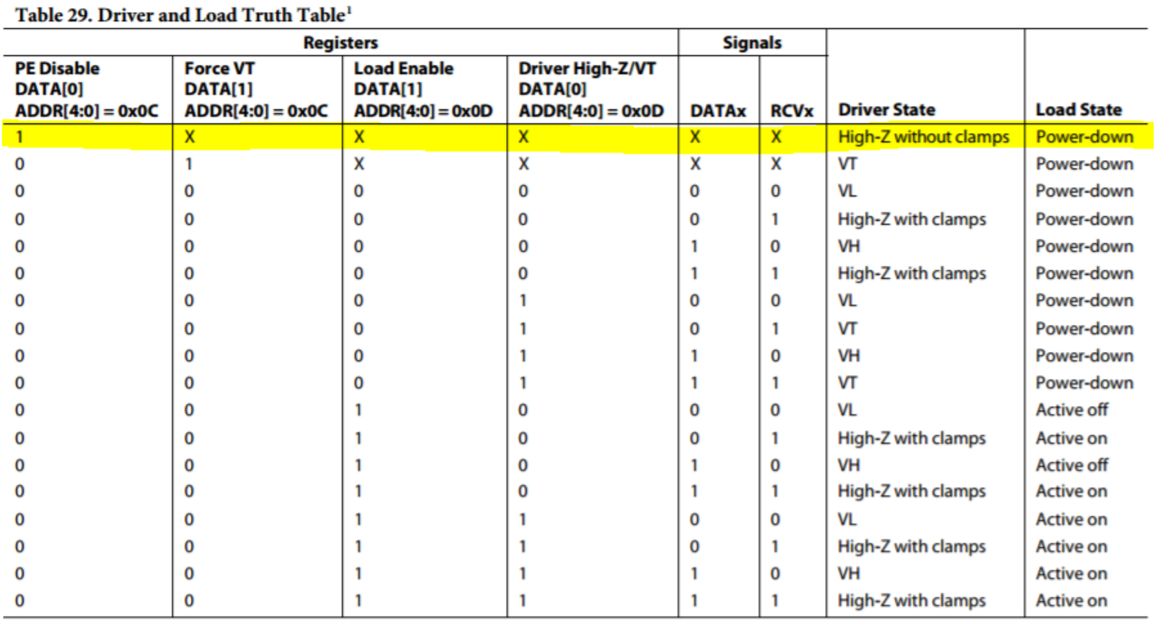


Table 29. Driver and Load Truth Table[1]

Registers				Signals			
PE Disable DATA[0] ADDR[4:0] = 0x0C	**Force VT DATA[1] ADDR[4:0] = 0x0C**	**Load Enable DATA[1] ADDR[4:0] = 0x0D**	**Driver High-Z/VT DATA[0] ADDR[4:0] = 0x0D**	**DATAx**	**RCVx**	**Driver State**	**Load State**
1	X	X	X	X	X	High-Z without clamps	Power-down
0	1	X	X	X	X	VT	Power-down
0	0	0	0	0	0	VL	Power-down
0	0	0	0	0	1	High-Z with clamps	Power-down
0	0	0	0	1	0	VH	Power-down
0	0	0	0	1	1	High-Z with clamps	Power-down
0	0	0	1	0	0	VL	Power-down
0	0	0	1	0	1	VT	Power-down
0	0	0	1	1	0	VH	Power-down
0	0	0	1	1	1	VT	Power-down
0	0	1	0	0	0	VL	Active off
0	0	1	0	0	1	High-Z with clamps	Active on
0	0	1	0	1	0	VH	Active off
0	0	1	0	1	1	High-Z with clamps	Active on
0	0	1	1	0	0	VL	Active on
0	0	1	1	0	1	High-Z with clamps	Active on
0	0	1	1	1	0	VH	Active on
0	0	1	1	1	1	High-Z with clamps	Active on

  ATE305 0x0C DATA[0] = 1, DATA[1] = 0, DATA[2] = 0

  ```
      W    0x38       0x1               // DATA[2:0] = 3’b001 for 0x0C
  ```

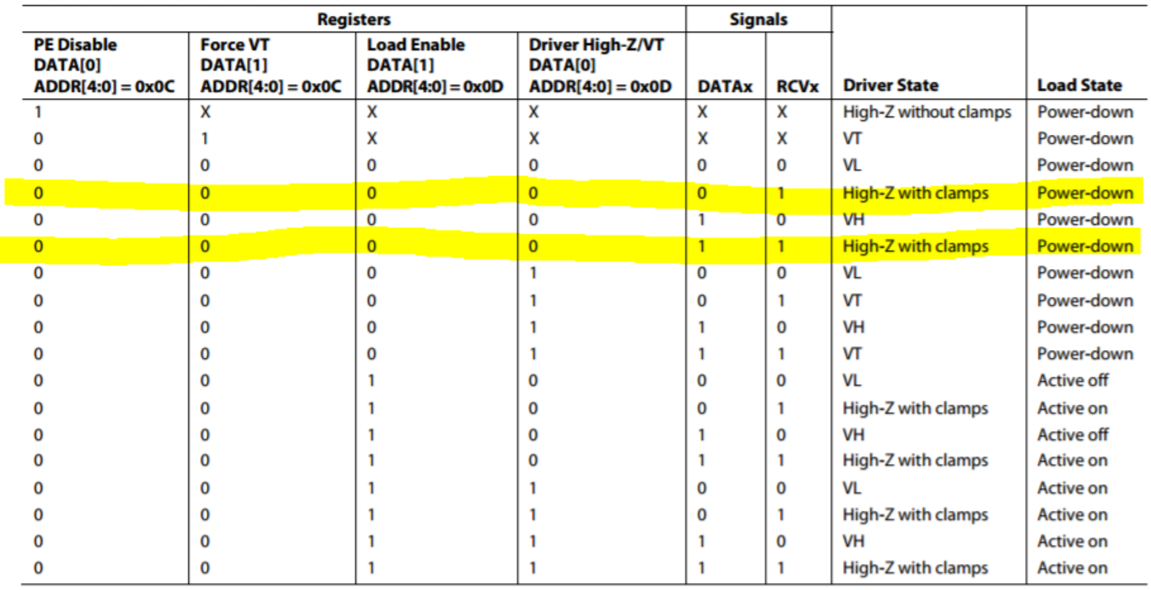
  ATE305_SEL = (0x01 << (CH_SEL / 2))

  ATE305_CH_SEL = CH_SEL % 2

  ```
      W    0x37       {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0C}    // Set 0x0C Register
  ```

針對 PMU Mode 我們需將指定的 Channel 開啟為 PMU，並因為要使用 Clamp 功能，使用的 Table 如下:

Table 29. Driver and Load Truth Table[1]



Registers				Signals			
PE Disable DATA[0] ADDR[4:0] = 0x0C	Force VT DATA[1] ADDR[4:0] = 0x0C	Load Enable DATA[1] ADDR[4:0] = 0x0D	Driver High-Z/VT DATA[0] ADDR[4:0] = 0x0D	DATAx	RCVx	Driver State	Load State
1	X	X	X	X	X	High-Z without clamps	Power-down
0	1	X	X	X	X	VT	Power-down
0	0	0	0	0	0	VL	Power-down
0	0	0	0	0	1	High-Z with clamps	Power-down
0	0	0	0	1	0	VH	Power-down
0	0	0	0	1	1	High-Z with clamps	Power-down
0	0	0	1	0	0	VL	Power-down
0	0	0	1	0	1	VT	Power-down
0	0	0	1	1	0	VH	Power-down
0	0	0	1	1	1	VT	Power-down
0	0	1	0	0	0	VL	Active off
0	0	1	0	0	1	High-Z with clamps	Active on
0	0	1	0	1	0	VH	Active off
0	0	1	0	1	1	High-Z with clamps	Active on
0	0	1	1	0	0	VL	Active on
0	0	1	1	0	1	High-Z with clamps	Active on
0	0	1	1	1	0	VH	Active on
0	0	1	1	1	1	High-Z with clamps	Active on

當使用 PMU 的時候，其他 Driver 應該為 High-Z 才不會影響 PMU Driver 的輸出或量測

PMU_MODE = PMU_MODE || (0x0001 << CH_SEL)

```
    W   0x3E    PMU_MODE   // Make RCV = 1, DATA don’t care
    W   0x3C    PMU_MODE   // Enable PMU Mode
```

ATE305 0x0C DATA[0] = 0, DATA[1] = 0, DATA[2] = 1

```
    W   0x38    0x4        // DATA[2:0] = 3’b100 for 0x0C
    W   0x37    {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0C}   // Set 0x0C Register
```

ATE305 0x0D DATA[0] = 0, DATA[1] = 0, DATA[2] = 0

```
    W   0x38    0x0        // DATA[2:0] = 3’b000 for 0x0D
    W   0x37    {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0D}   // Set 0x0D Register
```

針對 VTT Mode，跑 Pattern 時若 RCV= 1 則原本的 High-Z 要改為 terminate 至 VTT，我們使用 ATE305 spec Table29 如下:

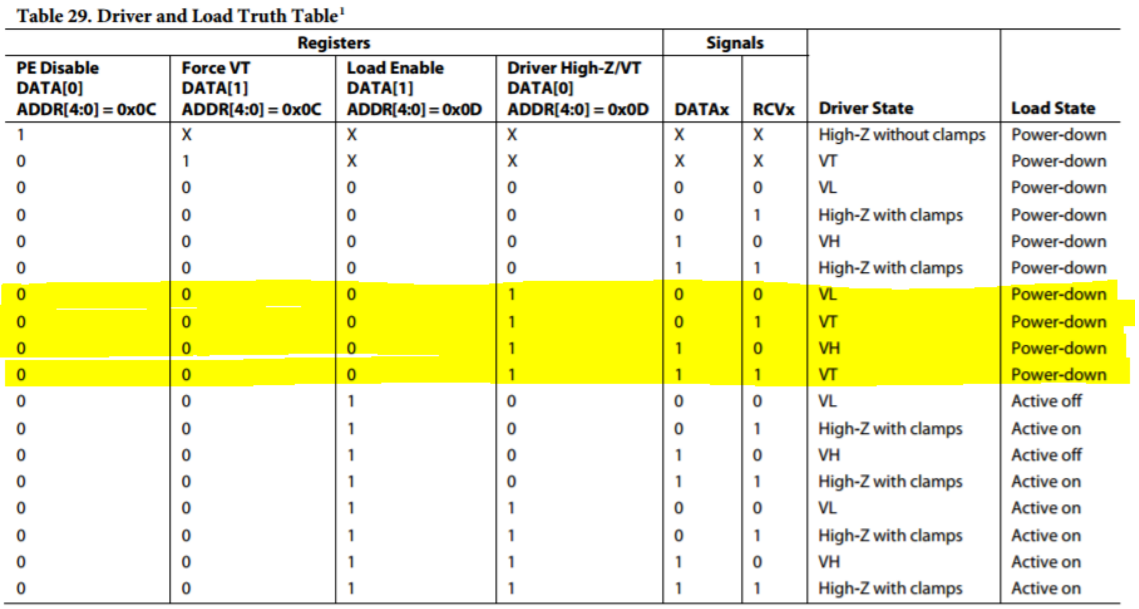


Table 29. Driver and Load Truth Table[1]

Registers				Signals			
PE Disable DATA[0] ADDR[4:0] = 0x0C	Force VT DATA[1] ADDR[4:0] = 0x0C	Load Enable DATA[1] ADDR[4:0] = 0x0D	Driver High-Z/VT DATA[0] ADDR[4:0] = 0x0D	DATAx	RCVx	Driver State	Load State
1	X	X	X	X	X	High-Z without clamps	Power-down
0	1	X	X	X	X	VT	Power-down
0	0	0	0	0	0	VL	Power-down
0	0	0	0	0	1	High-Z with clamps	Power-down
0	0	0	0	1	0	VH	Power-down
0	0	0	0	1	1	High-Z with clamps	Power-down
0	0	0	1	0	0	VL	Power-down
0	0	0	1	0	1	VT	Power-down
0	0	0	1	1	0	VH	Power-down
0	0	0	1	1	1	VT	Power-down
0	0	1	0	0	0	VL	Active off
0	0	1	0	0	1	High-Z with clamps	Active on
0	0	1	0	1	0	VH	Active off
0	0	1	0	1	1	High-Z with clamps	Active on
0	0	1	1	0	0	VL	Active on
0	0	1	1	0	1	High-Z with clamps	Active on
0	0	1	1	1	0	VH	Active on
0	0	1	1	1	1	High-Z with clamps	Active on

首先確定 Channel FPGA PMU 模式為關閉

PMU_MODE = PMU_MODE && (~(0x0001 << CH_SEL))

```
    W    0x3C      PMU_MODE    // Disable PMU Mode
```

ATE305 0x0C DATA[0] = 0, DATA[1] = 0, DATA[2] = 0

```
    W    0x38      0x0              // DATA[2:0] = 3’b000 for 0x0C
    W    0x37      {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0C}    // Set 0x0C Register
```

ATE305 0x0D DATA[0] = 1, DATA[1] = 0, DATA[2] = 0

```
    W    0x38      0x1              // DATA[2:0] = 3’b001 for 0x0D
    W    0x37      {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0D}    // Set 0x0D Register
```

針對 LOAD Mode 跑 Pattern 時若 RCV=0 則要開啟 Active Load，使用 Table 如下:

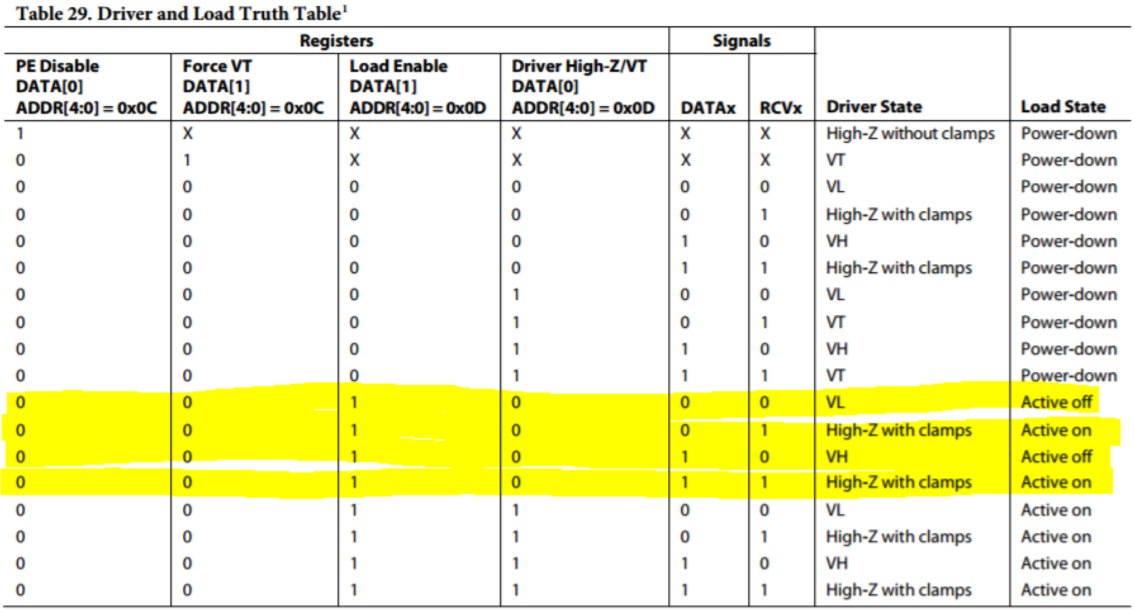


Table 29. Driver and Load Truth Table[1]

Registers				Signals			
PE Disable DATA[0] ADDR[4:0] = 0x0C	Force VT DATA[1] ADDR[4:0] = 0x0C	Load Enable DATA[1] ADDR[4:0] = 0x0D	Driver High-Z/VT DATA[0] ADDR[4:0] = 0x0D	DATAx	RCVx	Driver State	Load State
1	X	X	X	X	X	High-Z without clamps	Power-down
0	1	X	X	X	X	VT	Power-down
0	0	0	0	0	0	VL	Power-down
0	0	0	0	0	1	High-Z with clamps	Power-down
0	0	0	0	1	0	VH	Power-down
0	0	0	0	1	1	High-Z with clamps	Power-down
0	0	0	1	0	0	VL	Power-down
0	0	0	1	0	1	VT	Power-down
0	0	0	1	1	0	VH	Power-down
0	0	0	1	1	1	VT	Power-down
0	0	1	0	0	0	VL	Active off
0	0	1	0	0	1	High-Z with clamps	Active on
0	0	1	0	1	0	VH	Active off
0	0	1	0	1	1	High-Z with clamps	Active on
0	0	1	1	0	0	VL	Active on
0	0	1	1	0	1	High-Z with clamps	Active on
0	0	1	1	1	0	VH	Active on
0	0	1	1	1	1	High-Z with clamps	Active on

首先確定 Channel FPGA PMU 模式為關閉

```
PMU_MODE = PMU_MODE && (~(0x0001 << CH_SEL))
    W    0x3C      PMU_MODE    // Disable PMU Mode
```

ATE305 0x0C DATA[0] = 0, DATA[1] = 0, DATA[2] = 0

```
    W    0x38      0x0              // DATA[2:0] = 3'b000 for 0x0C
    W    0x37      {ATE305_SEL, 1'b1, (0x1 << ATE305_CH_SEL), 5'h0C}    // Set 0x0C Register
```

ATE305 0x0D DATA[0] = 0, DATA[1] = 1, DATA[2] = 0

```
    W    0x38      0x2              // DATA[2:0] = 3'b010 for 0x0D
    W    0x37      {ATE305_SEL, 1'b1, (0x1 << ATE305_CH_SEL), 5'h0D}    // Set 0x0D Register
```

針對 Normal Mode 跑 Pattern 時，沒有 load 也不會 terminate 至 VTT，使用 Table 如下:

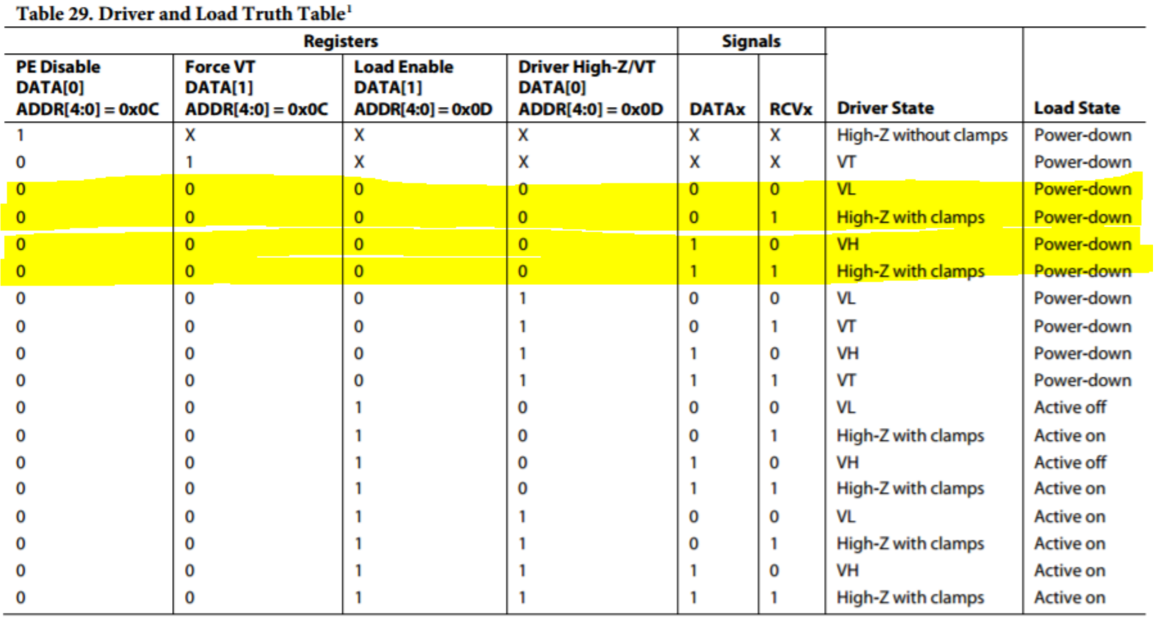


Table 29. Driver and Load Truth Table[1]

Registers				Signals			
PE Disable DATA[0] ADDR[4:0] = 0x0C	Force VT DATA[1] ADDR[4:0] = 0x0C	Load Enable DATA[1] ADDR[4:0] = 0x0D	Driver High-Z/VT DATA[0] ADDR[4:0] = 0x0D	DATAx	RCVx	Driver State	Load State
1	X	X	X	X	X	High-Z without clamps	Power-down
0	1	X	X	X	X	VT	Power-down
0	0	0	0	0	0	VL	Power-down
0	0	0	0	0	1	High-Z with clamps	Power-down
0	0	0	0	1	0	VH	Power-down
0	0	0	0	1	1	High-Z with clamps	Power-down
0	0	0	1	0	0	VL	Power-down
0	0	0	1	0	1	VT	Power-down
0	0	0	1	1	0	VH	Power-down
0	0	0	1	1	1	VT	Power-down
0	0	1	0	0	0	VL	Active off
0	0	1	0	0	1	High-Z with clamps	Active on
0	0	1	0	1	0	VH	Active off
0	0	1	0	1	1	High-Z with clamps	Active on
0	0	1	1	0	0	VL	Active on
0	0	1	1	0	1	High-Z with clamps	Active on
0	0	1	1	1	0	VH	Active on
0	0	1	1	1	1	High-Z with clamps	Active on

首先確定 Channel FPGA PMU 模式為關閉

PMU_MODE = PMU_MODE && (~(0x0001 << CH_SEL))

```
    W   0x3C    PMU_MODE   // Disable PMU Mode
```

ATE305 0x0C DATA[0] = 0, DATA[1] = 0, DATA[2] = 0

```
    W   0x38    0x0            // DATA[2:0] = 3'b000 for 0x0C
    W   0x37    {ATE305_SEL, 1'b1, (0x1 << ATE305_CH_SEL), 5'h0C}   // Set 0x0C Register
```

ATE305 0x0D DATA[0] = 0, DATA[1] = 0, DATA[2] = 0

```
    W   0x38    0x0            // DATA[2:0] = 3'b000 for 0x0D
    W   0x37    {ATE305_SEL, 1'b1, (0x1 << ATE305_CH_SEL), 5'h0D}   // Set 0x0D Register
```

上面的 Pin_Mode API 若將某一 Channel 的 Pin_Mode 設定為”PMU”，則可以使用下面 API:

- FV (channel, Value, I_range, clamp_H, clamp_L, delay)   // default delay = 0ms
- FI (channel, Value, I_range, clamp_H, clamp_L, delay)   // default delay = 0ms
- MI (channel, count)   // default count(1)

上面的 Pin_Mode API 除”OFF”模式之外，都可以使用 MV:

- MV (channel, count)   // default count(1)

FV API 底層撰寫指導:

FV (channel, Value, I_range, clamp_H, clamp_L, delay)

由於 FV 只能在 PMU 模式下使用，所以首先要設定為 PMU 模式:

Pin_Mode(PMU)

將 ClampH/L 的值先設定於 ATE305

```
W    0x38     ClampH
W    0x37     {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h06}   // Set 0x06 Register
W    0x38     ClampL
W    0x37     {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h07}   // Set 0x07 Register
```

Force Voltage 的來源為 PMUDAC，以輸入參數”Value”代入並運算公式得 FV_VALUE

```
W    0x38     FV_VALUE
W    0x37     {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0B}   // Set 0x0B Register
```

ATE305 0x0E DATA[2:0] = I_range, DATA[3] = 0, DATA[4] = 1, DATA[5] = 1, DATA[7] = 0, DATA[9:8] = 2’b11

```
W    0x38     DATA
W    0x37     {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0E}   // Set 0x0E Register
```

FI API 底層撰寫指導:

FI (channel, Value, I_range, clamp_H, clamp_L, delay)

由於 FI 只能在 PMU 模式下使用，所以首先要設定為 PMU 模式:

Pin_Mode(PMU)

將 ClampH/L 的值先設定於 ATE305

```
W    0x38     ClampH
W    0x37     {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h06}   // Set 0x06 Register
W    0x38     ClampL
W    0x37     {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h07}   // Set 0x07 Register
```

Force Current 的來源為 PMUDAC，以輸入參數”Value、I_range”代入並運算公式得 FI_VALUE

```
W    0x38     FI_VALUE
W    0x37     {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0B}   // Set 0x0B Register
```

ATE305 0x0E DATA[2:0] = I_range, DATA[3] = 1, DATA[4] = 0, DATA[5] = 1, DATA[7] = 0, DATA[9:8] = 2’b11

```
W    0x38     DATA
W    0x37     {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0E}   // Set 0x0E Register
```

MI API 底層撰寫指導:

MI (channel, count)

由於 MI 只能在 PMU 模式下使用，所以首先要設定為 PMU 模式:

```
Pin_Mode(PMU)
```

在 MI 之前不知道是 FV 或 FI，故先回讀 ATE305 0x0E 當下資料

```
W   0x37      {ATE305_SEL, 1’b0, (0x1 << ATE305_CH_SEL), 5’h0E}   // Read 0x0E Register
R   0x38      READ_DATA
```

將回讀回來的 0x0E 狀態改為 measure current mode

```
DATA = READ_DATA || 9’b0000010000
```

處理過後寫回 0x0E

```
W   0x38      DATA
W   0x37      {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0E}   // Set 0x0E Register
```

開啟量測通道

```
W   0x38      {1’b0, ATE305_CH_SEL, 1’b1}
W   0x37      {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0F}   // Set 0x0F Register
```

固定 delay 一段時間

```
Delay();
```

發出 PMU ADC 量測命令

```
W   0x39      count
```

確認量測完成

```
R   0x39      WORKING = 0
```

讀取結果

```
R   0x3A      ADC_MSB
R   0x3B      ADC_LSB
```

依照一開始讀回的 READ_DATA[2:0]決定 range 選擇公式，並回傳結果

關閉量測通道，避免通道衝突

```
W   0x38      0x0
W   0x37      {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0F}   // Set 0x0F Register
```

MV API 底層撰寫指導:

MV (channel, count)

MV 除了 OFF 模式之外都能使用，但在 MV 之前不知道是否有 FV 或 FI，故先回讀 ATE305 0x0E 當下資料

```
W    0x37      {ATE305_SEL, 1’b0, (0x1 << ATE305_CH_SEL), 5’h0E}    // Read 0x0E Register
R    0x38      READ_DATA
```

將回讀回來的 0x0E 狀態改為 measure voltage mode

```
DATA = READ_DATA && 9’b1111101111
```

處理過後寫回 0x0E

```
W    0x38      DATA
W    0x37      {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0E}    // Set 0x0E Register
```

開啟量測通道

```
W    0x38      {1’b0, ATE305_CH_SEL, 1’b1}
W    0x37      {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0F}    // Set 0x0F Register
```

固定 delay 一段時間

```
Delay();
```

發出 PMU ADC 量測命令

```
W    0x39      count
```

確認量測完成

```
R    0x39      WORKING = 0
```

讀取結果

```
R    0x3A      ADC_MSB
R    0x3B      ADC_LSB
```

MV 只有一個公式，運算後回傳結果

關閉量測通道，避免通道衝突

```
W    0x38      0x0
W    0x37      {ATE305_SEL, 1’b1, (0x1 << ATE305_CH_SEL), 5’h0F}    // Set 0x0F Register
```