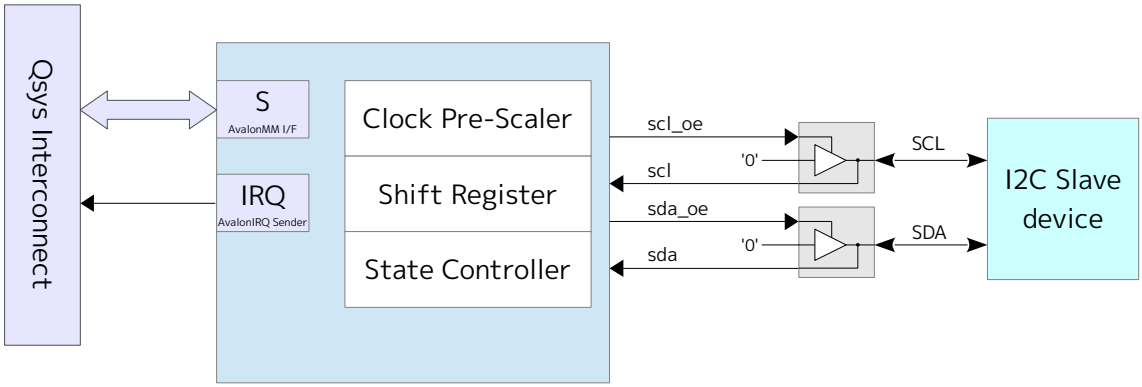
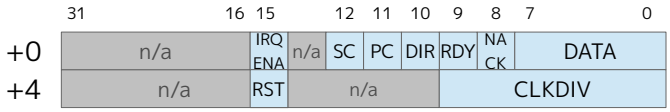
PERIDOT I2Cマスタ仕様書

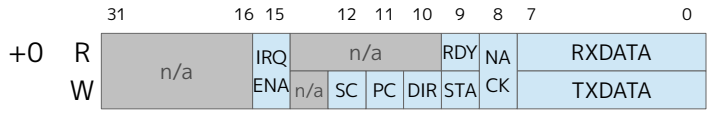
●全体ブロック図



●レジスタマップ



・I2Cアクセスレジスタ



IRQENA - 割り込み有効レジスタ
I2Cバスの通信が完了したときに割り込みを発行する。

'0' : 割り込み無効 ※初期値
'1' : 割り込み有効

RDYが'1'の時に割り込みが発生するため、STAの書き込みと同時に有効にしなければならない。

RDY - ペリフェラルレディレジスタ
このレジスタが'0'の時はアクセス実行中またはペリフェラルリセット状態で、RSTレジスタを除く全てのレジスタへの書き込みがブロックされる。

STA - アクセススタートレジスタ
RDYが'1'の時にこのレジスタへ'1'を書き込むとI2Cバスの通信を開始する。

SC – スタートコンディション発行レジスタ
このフィールドに'1'が設定されている場合、I2Cバイトアクセスの直前にスタートコンディションを発行する。先行のアクセスがI2Cバイトアクセスの場合、リピートスタートコンディションを発行する。

PC - ストップコンディション発行レジスタ
このフィールドに'1'が設定されている場合、I2Cバイトアクセスの直後にストップコンディションを発行する。

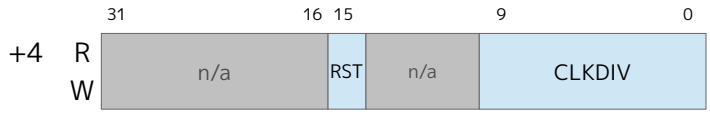
DIR – リードライト設定レジスタ
このフィールドに'1'が設定されてる場合、I2Cバイトアクセスはリード、'0'の場合はライトとなる。

NACK – アクノリッジレジスタ
リード(DIR='1')時はこのレジスタに設定された値をACK応答としてデバイスに送信する。ライト(DIR='0')時はデバイスからのACK応答がこのレジスタに格納される。

RXDATA – 受信データバイトレジスタ
受信したデータバイトを読み出すレジスタ。リードアクセス発行後、RDYが'1'の時に有効な値を返す。

TXDATA - 送信データバイトレジスタ
送信するデータバイトを書き込むレジスタ。ライトアクセス発行時にこのフィールドの値を取り込む。

・I2C設定レジスタ



RST - ペリフェラルリセットレジスタ
I2Cペリフェラルのリセットを行うレジスタ。リセット状態では他の全てのレジスタへのアクセスがブロックされる。

'0' : リセット解除
'1' : ペリフェラルリセット ※初期値

CLKDIV - プリスケーラレジスタ
I2C通信のクロック速度を設定するプリスケーラ。通信速度は次の式により決定する。

bitrate[bps] = clock[Hz] / ((CLKDIV + 5) × 4)

●補足

I2Cマスタの使用手順例を下記に示す。特に注記の無い場合、ペリフェラル名を"I2C"、動作クロックを25MHzでインスタンスしたものとする。

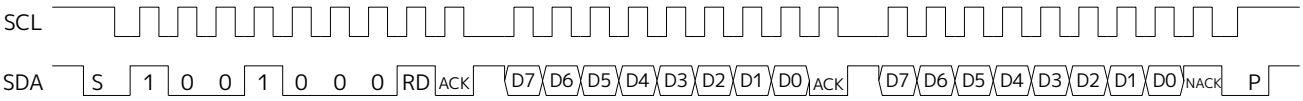
・ペリフェラル初期化

（１）RSTレジスタをクリアする
（２）RDYが'1'になるのを待つ
（３）CLKDIVに分周値をセットする（100kbps＝58、400kbps＝11）

・スレーブアドレス0x48から2バイトリード

（１）レジスタ０に0x1291（SC='1', STA='1', TXDATA=0x91）を書き込む
（２）RDYが'1'になるのを待つ
（３）スレーブが応答すればNACKレジスタに'0'が返る
（４）レジスタ０に0x600（DIR='1', STA='1', NACK='0'）を書き込む
（５）RDYが'1'になるのを待つ
（６）RXDATAレジスタから１バイト目のデータを取得
（７）レジスタ０に0xF00（PC='1', DIR='1', STA='1', NACK='1'）を書き込む
（８）RDYが'1'になるのを待つ
（９）RXDATAレジスタから２バイト目のデータを取得

動作波形



・Cソース例

```
// 初期化

IOWR(I2C_BASE, 1, 0);
while( !((IORD(I2C_BASE, 0) & (1<<9)) ){}
IOWR(I2C_BASE, 1, 58);

// スレーブ0x48から16bitデータをリード

IOWR(I2C_BASE, 0, (1<<12)|(1<<9)|(0x48<<1)|1);
while( !((IORD(I2C_BASE, 0) & (1<<9)) ){}
if ( IORD(I2C_BASE, 0) & (1<<8) ) return -1;

IOWR(I2C_BASE, 0, (1<<10)|(1<<9));
while( !((IORD(I2C_BASE, 0) & (1<<9)) ){}
data = IORD(I2C_BASE, 0) & 255;

IOWR(I2C_BASE, 0, (1<<11)|(1<<10)|(1<<9)|(1<<8));
while( !((IORD(I2C_BASE, 0) & (1<<9)) ){}
data = (data<<8) | IORD(I2C_BASE, 0) & 255;

return data;
```

PERIDOT I2Cマスタ仕様書

●変更履歴

2017/02/20
　　Rev.2リリース / s.osafune
　　16.1対応およびバージョン名の変更
2015/05/26
　　Rev.1リリース / s.osafune