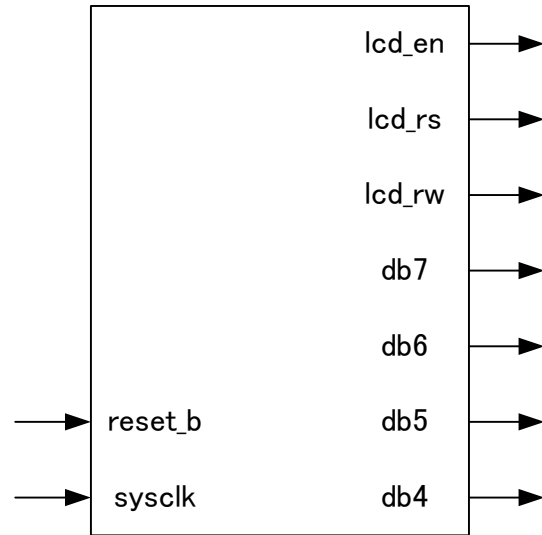


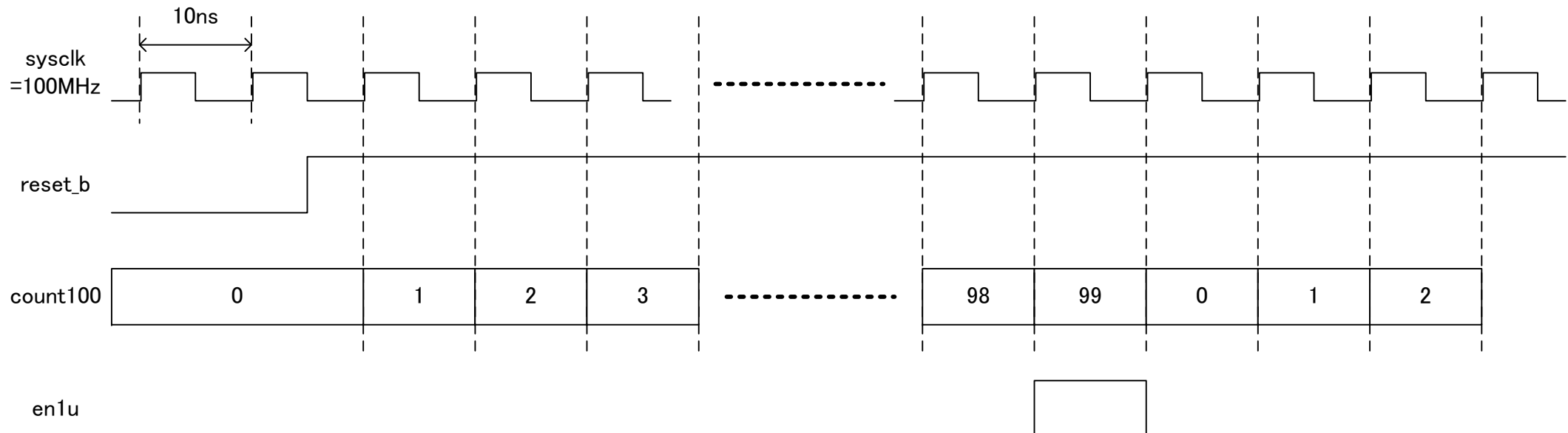
Entity: LCD_DRV0



STEP0

左図に示されるENTITY回路を設計する。
STEP0では、ENTITYの出力信号7本はまだ設計しないので、すべての出力信号を'0'としておく。

まず、最初に、リセット信号解除後(HからL変化後)に、0から99まで数えるカウンタを設計する。99の次は、0に戻り、同じ動作を繰り返す。
カウンタの値が99の時にだけ、HIGHとなるパルス信号en1uを生成する。



STEP1

STEP0の設計結果、en1uという1usごとにパルス生成をする信号ができた。

このSTEP1では、このen1uのパルスを数えるカウンタを設計する。このパルスを0から1999まで数えて、1999の次は0に戻るとする。

カウント値が1999の時に、en2mなる信号にパルスを生成する。

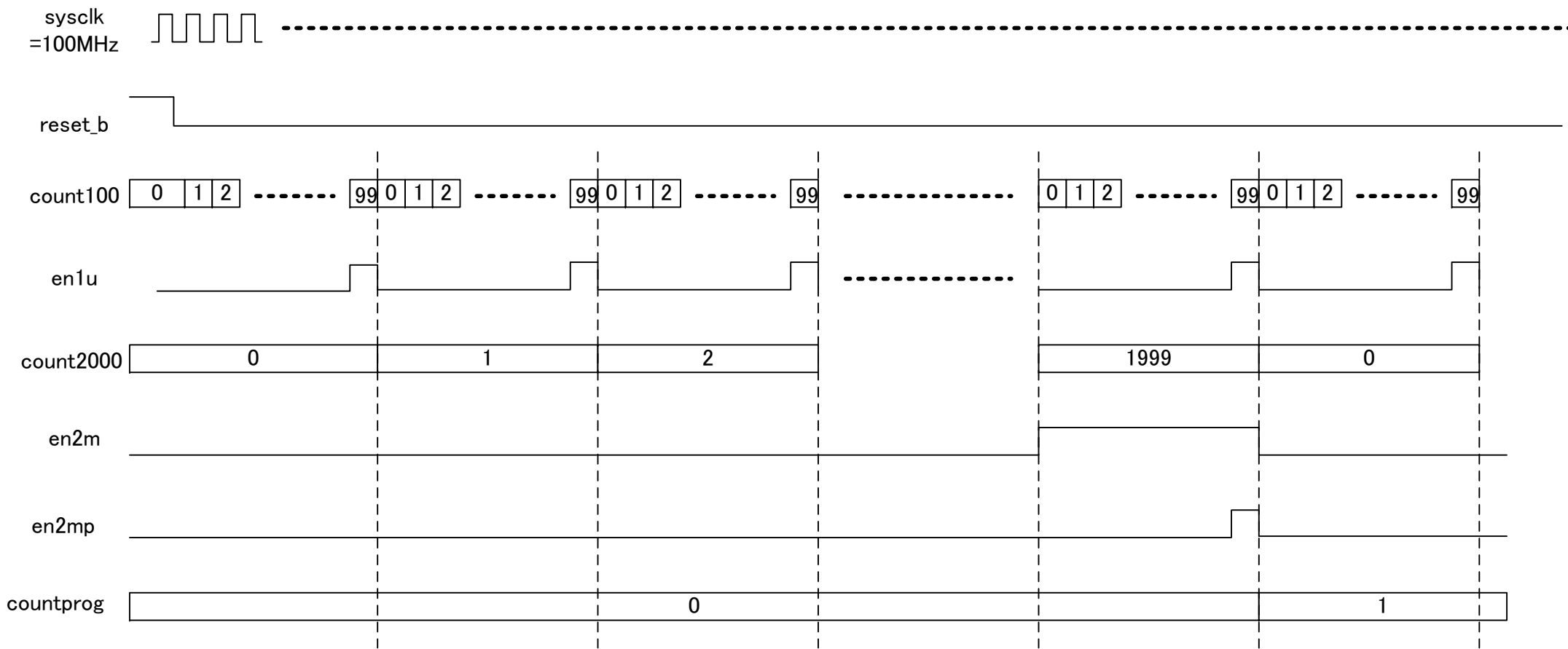
そして、en2mとen1uのANDを取った信号をen2mpとする。このen2mpはパルス幅が1クロックサイクルに信号である。

そして、このen2mpパルスごとにカウントアップするcountprogを追加する。このcountprogは0から63すなわちを繰り返す。すなわち、6ビットのカウンタとなる。

これまでの設計で、1マイクロ秒(1us)ごとに生成されるパルスen1u信号と、2ミリ秒(2ms)ごとに生成されるパルスen2mp信号が生成された。

実習の最後には、en2mpパルスをトリガとして、LCDコントローラに命令やデータを送る回路を設計する。

LCDコントローラの動作は遅いので、1usのen1u信号を用いて、すなわち1usごとに必要な波形を入力する。



STEP2

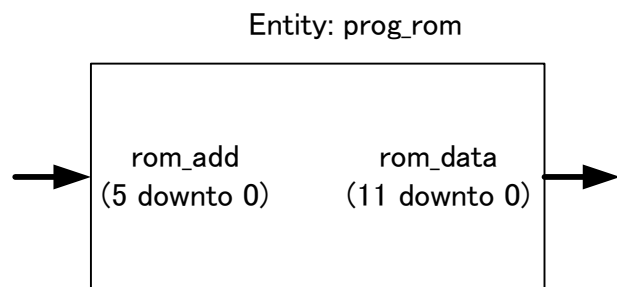
ROM設計

LCDコントローラへの転送する一連の命令をROMデータとして蓄えておきます。

ROMの内容を順に読み出し、必要な波形を生成し、LCDコントローラへ必要命令やデータを転送するためです。

これまでの、STEP0,STEP1ではFFを生成する順序回路ですが、このROMは組み合わせ回路として設計します。

ROMのサイズは64ワードX12ビットですので、アドレス入力信号は6ビット、データ出力信号は12ビットとなります。



	NEXT	GO	CYC	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	説明
	rom_data												
rom_add	<11>	<10>	<9>	<8>	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>	
0	1	0	0	0	0	0	0	0	0	0	0	0	GO=0なので、何もしないNOP
1	1	1	0	0	0	0	1	1	0	0	0	0	Function Set
2	1	1	1	0	0	0	1	0	0	0	0	0	Function Set
3	1	1	1	0	0	0	1	0	0	0	0	0	Function Set
4	1	1	1	0	0	0	0	0	1	1	0	0	Display ON
5	1	1	1	0	0	0	0	0	0	0	0	1	Display Clear
6	1	1	1	0	0	0	0	0	0	1	1	0	Entry Mode Set
7	1	1	1	1	0	1	0	0	0	0	0	1	A'を表示
8	0	0	0	0	0	0	0	0	0	0	0	0	NEXT=0でSTOP,GO=0でNOP
9	以下はALL'0'												
10													
.													
.													
.													
.													
.													
.													
.													
63													

上記ROMが設計できたら、STEP1で設計した回路に組み込み、countprogなる6ビット信号をrom_addに入力することで、シミュレーションでROMを読み出し、ROMの内容を確認せよ。

STEP3

ROMの読み出しデータrom_dataのビット<11>はNEXT信号と呼び、'1'の時は、countprogが'1'上昇し、'0'の時は、countprogが固定する。すなわち、同じROMアドレスを読み続けることになり、STOP状態となる

くりかえすが、

NEXT=1の時は、countprogを+1する、0のときはcountprogを変化させないように、STEP3で設計した、countprog生成回路を修正せよ。

STEP4

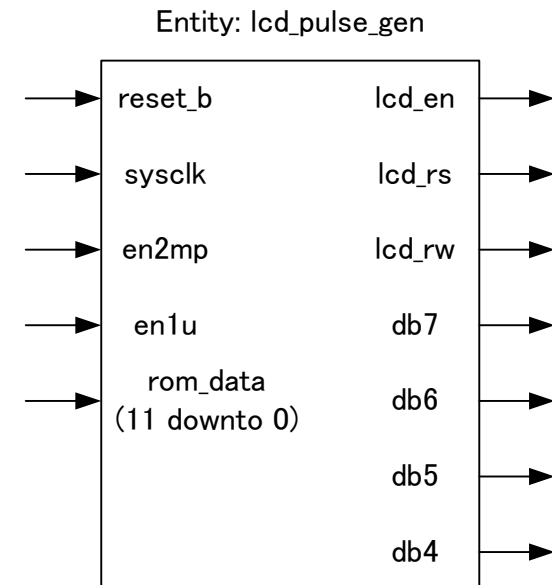
右図でしめされるentity: lcd_pulse_gen を設計する。

次ページに、動作波形を示す。

en2mpが'1'の時、読みだされたrom_dataの値によって、LCDに関する信号を生成する。

- ①GO=1の時はパルス生成するが、0の時は動作しない。
- ②CYC=0の時は内部カウンタの最低値を3とし、5→4→3の3サイクル動作をし、CYC=1の時は、内部カウンタの最低値を0とし、6サイクル動作をする。
- ③lcd_rsには保持したデータ<8>を出力する。Lcd_rwはつねに'0'を出力する。
- ④内部カウンタが5,4,3の場合は、db<7,6,5,4>=保持したデータの<7,6,5,4>とし、内部カウンタが2,1,0の場合は、db<7,6,5,4>=保持したデータの<3,2,1,0>とする。
- ⑤内部カウンタが4, 1の時にlcd_en=1とし、それ以外は'0'とする。

この設計された、lcd_pulse_genをSTEP3の回路に組み込み、lcd_en等のLCD制御信号を生成させる。



STEP4

en2mpが'1'の時、読みだされたrom_dataの値によって、LCDに関する信号を生成する。

- ①GO=1の時はパルス生成するが、0の時は動作しない。
- ②CYC=0の時は内部カウンタの最低値を3とし、5→4→3の3サイクル動作をし、CYC=1の時は、内部カウンタの最低値を0とし、6サイクル動作をする。
- ③lcd_rsには保持したデータ<8>を出力する。Lcd_rwはつねに'0'を出力する。
- ④内部カウンタが5,4,3の場合は、db<7,6,5,4>=保持したデータの<7,6,5,4>とし、内部カウンタが2,1,0の場合は、db<7,6,5,4>=保持したデータの<3,2,1,0>とする。
- ⑤内部カウンタが4, 1の時にlcd_en=1とし、それ以外は'0'とする。



STEP5

以下に示される、FPGAのピンとの接続等を示したUCFファイルを組み込み、回路合成し、FPGAダウンロードし、

LCDディスプレイに A の文字が表示されることを確認せよ。

```
## LCD ##
Net db4 LOC=AB10 | IOSTANDARD = LVCMOS33 | PULLUP | SLEW = SLOW | DRIVE = 2 | TIG;
Net db5 LOC=AC10 | IOSTANDARD = LVCMOS33 | PULLUP | SLEW = SLOW | DRIVE = 2 | TIG;
Net db6 LOC=AE12 | IOSTANDARD = LVCMOS33 | PULLUP | SLEW = SLOW | DRIVE = 2 | TIG;
Net db7 LOC=AF12 | IOSTANDARD = LVCMOS33 | PULLUP | SLEW = SLOW | DRIVE = 2 | TIG;
Net lcd_rw LOC=AB17 | IOSTANDARD = LVCMOS33 | PULLUP | SLEW = SLOW | DRIVE = 2 | TIG;
Net lcd_rs LOC=AC17 | IOSTANDARD = LVCMOS33 | PULLUP | SLEW = SLOW | DRIVE = 2 | TIG;
Net lcd_en LOC=AE13 | IOSTANDARD = LVCMOS33 | PULLUP | SLEW = SLOW | DRIVE = 2 | TIG;

Net sysclk TNM_NET = sys_clk_pin;
TIMESPEC TS_sys_clk_pin = PERIOD sys_clk_pin 100000 kHz;
Net sysclk LOC=AE14 | IOSTANDARD = LVCMOS33;
Net reset_b TIG;
Net reset_b LOC=D6 | PULLUP;
```