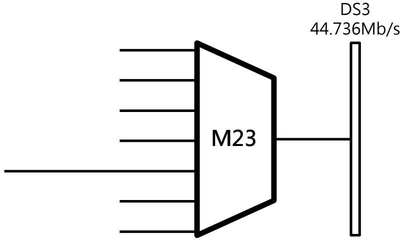
傳輸系統電路設計與模擬 - Mapper IC M23設計



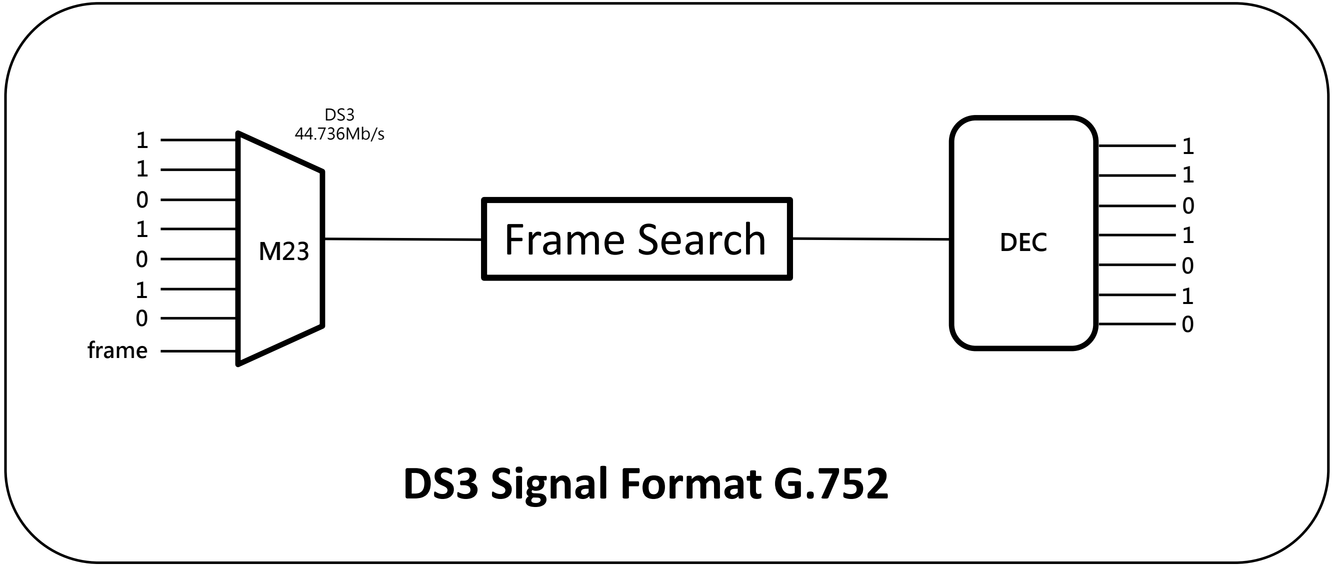
DS3 Signal Format G.752

教授: 郭昭宗

學生: 電子三乙 110510216 蔡承宏
電子三甲 110510134 王維碩

2019/6/11

功能概要



DS3 Signal Format G.752

Data: 85bits x 8 = 680 bits
85 clock one hadder
The full of M23 has 680bits x 7 =4760 bits

RX

接收端

程式介紹 - 初值設定

```
module DS3
(clk,reset,count,data,sel,in,pulse,subframe,f0,f1,F0,F1,
all,newframe,newpulse,one,two,three,four,five,six,seven,up);

input clk,reset;
output [9:0] count;
reg [9:0] count;

output [6:0] data;
reg [6:0] data;
output [3:0] sel;
reg [3:0] sel;
output in;
reg in;
output pulse;
reg pulse;
```

```
output subframe,f0,f1,F0,F1;
reg subframe,f0,f1,F0,F1;

output [12:0] all;
reg [12:0] all;
output [2:0] newframe;
reg [2:0] newframe;
output
newpulse,one,two,three,four,five,six,seven,up;
reg newpulse,one,two,three,four,five,six,seven,up;
```

程式介紹

```
// upcounter

always@(posedge clk)
begin
 if(reset)
    begin
       count=0; sel=0;
    end
 else if (count==679)
       count=0;
 else
       count = count + 1;
end
```

- 這裡利用正緣 clk 作為觸發
- 將選擇線還有計數器初值設為 0
- 讓計數器數 0~679 共 680 bits

```
// multiplexer upcounter
always@(posedge clk)
begin
  if (reset)
      sel=0;
  else if (sel==6)
      sel=0;
  else
      sel=sel+1;
end
```

- sel 用來控制 in ，輸入有7組，所以數 0 到 6
- 這邊讓它不斷的數 0 到 6 全部需要共有 4760 bits
- 達到輸入資料1101010 ，7組資料，用7條控制線控制

```
// multiplexer give frame seat
always@(count)
begin
      if (count==85 || count ==0)      sel=0;
 else if (count==170 || count==255 || count == 340 || count ==425 || count == 510 || count == 595 || count ==679)
          sel=0;
end
```

- 每經過84個bits會有一個資料對齊位元 第一個為位於count = 85 的 F1
- 於第一列中F1之後會遇到 C11、F0、C12、F0、C13、F1、還有最後位元 count = 679
- 這些時候皆須將sel歸零

```
// multiplexer
always@(sel)
begin
        if (sel==0)  in=1;
  else if (sel==1)  in=1;
  else if (sel==2)  in=0;
  else if (sel==3)  in=1;
  else if (sel==4)  in=0;
  else if (sel==5)  in=1;
  else if (sel==6)  in=0;
end

// data SISO
always@(posedge clk)
begin
  data[6] = data [5] ;
  data[5] = data [4] ;
  data[4] = data [3] ;
  data[3] = data [2] ;
  data[2] = data [1] ;
  data[1] = data [0] ;
  data[0] = in ;
end
```

- 當控制線為0 到 6 時分別對應其輸入
- 0 到 6 不斷重複輸入 1101010 當成資料
- data[ ]用來一個一個的導入 in 的值

```
// frame's seat
always@(posedge clk)
begin
  if (reset)
      pulse = 1;
  else if (count==85 || count==170 || count==255 || count == 340 || count ==425 || count == 510 || count == 595 || count ==0)
      pulse = 1;
  else
      pulse = 0;
end
```

- 以pulse作為識別開頭到結尾所遇到的這些對齊符號
- 符號以第1列舉例為 X1、F1、 C11、F0、C12、F0、C13、F1

```
// the first frame's number

always@(pulse)
begin
  if (pulse==1 && count ==0 && all! =2040 && all! =2720 && all! = 3400 )
      data=1;
end
```

- All 為7列 0~679 bits 的 資料
- 共 4760 bits，設定訊框開頭字元
- P2: count = 2040
- M0: count = 2720
- M1: count = 3400

```
// when the frame is 0 or 1
always@(clk)
begin
if (count ==170 || count ==255 || count==340 || count ==425 || count==510)
     data = 7'b1010100;
else if (count == 85 || count == 595 )
     data = 7'b1010101;
else if (all==3400  || all==2040)
     data = 7'b1010100;
else if (all==2720)
     data = 7'b1010101;
end
```

- 使用count 控制 data
- 將其指定於題目所規定的值

```
always@(count)
begin
  if (reset)
  begin  subframe =0;
            f0=0;  f1=0;  F0=0;  F1=0;
  end

  else if (count==85)
     F1 = 1;
  else if (count==255)
     F0 = 0;
  else if (count==425)
     f0 = 0;
  else if (count==595)
     f1 = 1;

  else
  begin  F1 = F1; F0 = F0; f0 = f0; f1 = f1;  end

end
```

- 將訊框預設值和偵測線於初始值時設為零

- 標示出Frame count = 85　的值
- 標示出Frame count = 255 的值
- 標示出Frame count = 425 的值
- 標示出Frame count = 595 的值

- 將值等於自己，避免在不等於自己時亂跑

```
always@(clk or f1)
begin
 if (F1==1 && F0 ==0 &&f0==0 && f1==1)
   subframe=1;
else if (all==4759)
 begin
  F1=0;
  f1=0;
  subframe=0;
 end
end

// when the frame is 0 or 1
```

```
always@(posedge clk)
begin
 if (reset)
  begin
    all=0;
  end
 else if (all==4759)
  begin
     all=0; F1=0; f1=0;  subframe=0;
  end
 else
     all=all+1;
end
```

- 偵測到規定的訊框值1001
- 使標示用 subframe設為1作為標示
- 經過0~4759bits 共4760bits結束第1次循環
- 並歸零

- 到 4759 bits 時歸零
- 並以 all=all+1 繼續循環

```
always@(posedge data)
begin
  if (reset)
  newframe=0;
  else if (count==0)
  begin
  newframe[2] = newframe[1];
  newframe[1] = newframe[0];
  newframe[0] = data[0];
  end
end

always@(newframe)
begin
  if (reset)
  newpulse  = 0;
  else if (newframe==3'b010)
  newpulse  = 1;
  else
  newpulse  = 0;
end
```

- 利用data[ ]收到的資料傳送至newframe[2:0] 3bits 之位元也就是7’ bxxxxxxx to 3’ bxxx

- 這邊利用newframe來偵測末三列的M0、M1、M0

```
//pulse
always@(posedge clk)
begin
  if (reset)
  begin
  one=0; two=0; three=0; four=0; five=0; six=0; seven=0;
  end
  else if (all<=679)
  one=1;
  else
  one=0;
end

always@(posedge clk)
begin
  if (all>=680 && all<=1359)
  two=1;
  else
  two=0;
end

always@(posedge clk)
begin
  if(all>=1360 && all<=2039)
  three=1;
  else
  three=0;
end
```

- 對於 7 條線做初值設 0 動作
- 識別 0 ~ 679 共680bits的範圍
- 表示第一列 X1 ~ 83I

- 識別 680 ~ 1359 bits為第二列
- 範圍 X2 ~ 82I

- 識別 1360 ~ 2039 bits為第三列
- 範圍 P1 ~ 81I

```
always@(posedge clk)
begin
  if  (all>=2040 && all<=2719)
  four=1;
  else
  four=0;
end

always@(posedge clk)
begin
  if (all>=2720 && all<=3399)
  five=1;
  else
  five=0;
end

always@(posedge clk)
begin
  if (all>=3400 && all<=4079)
  six=1;
  else
  six=0;
end
```

- 識別 2040 ~ 2719 bits為第四列
- 範圍 P2 ~ 80I

- 識別 2720 ~ 3399 bits為第五列
- 範圍 M0 ~ 79I

- 識別 3400 ~ 4079 bits為第六列
- 範圍 M1 ~ 78I

```
always@(posedge clk)
begin
  if (all>=4080 && all<=4759)
  seven=1;
  else
  seven=0;
end
//pulse
```

- 識別 4080 ~ 4759 bits為第六列
- 範圍 M0 ~ 77I

```
always@(posedge clk)
begin
  if (reset)
  up=0;
  else if (count==679)
  up=1;
  else
  up=0;
end


endmodule
```

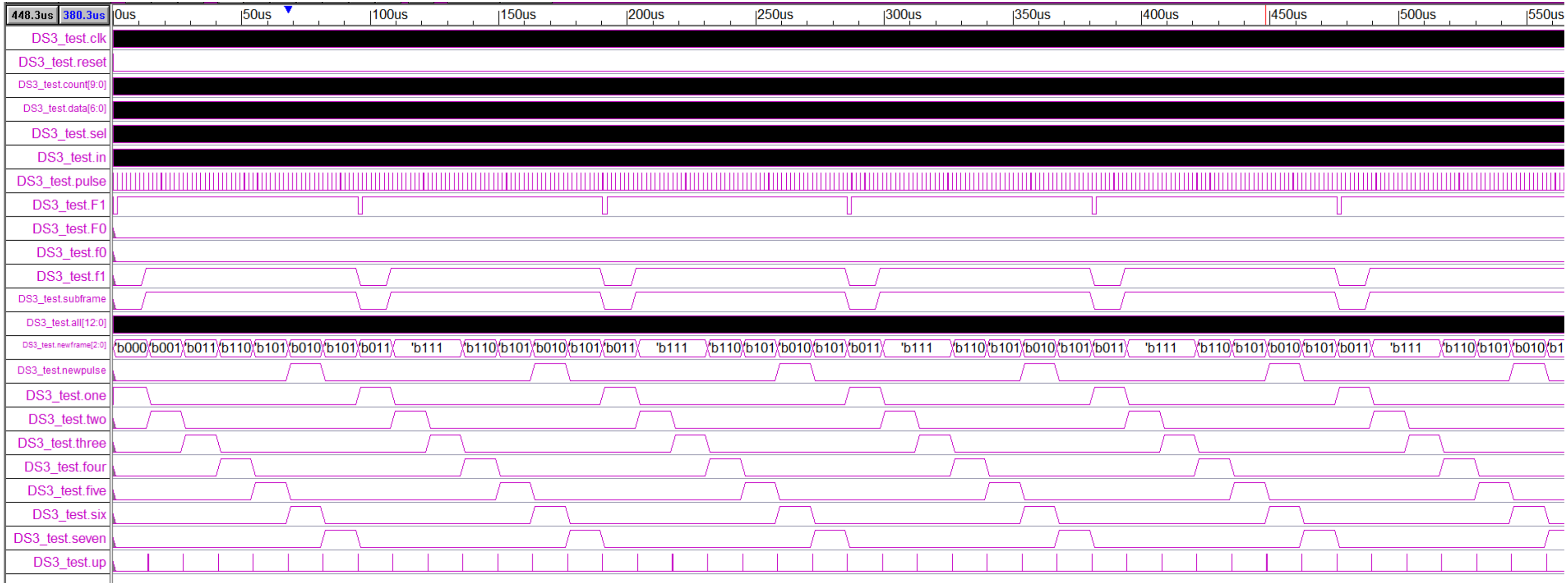
- 每 680 bits 設置一個up用來表示有偵測到一列
- 可將up波型與各列波型做對照確認

測試檔

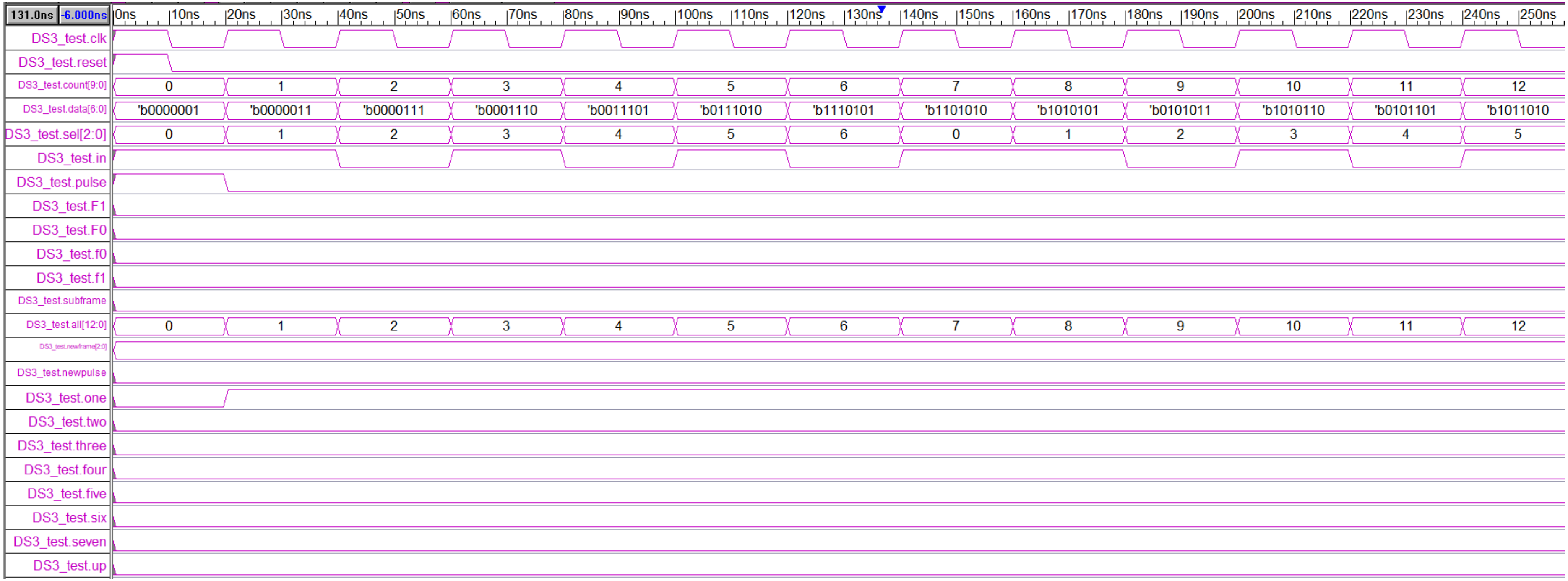
```
module DS3_test;
reg clk,reset;
wire [9:0] count;
wire [6:0] data;
wire [3:0] sel;
wire in,pulse,subframe,f0,f1,F0,F1;
wire [12:0] all;
wire [2:0] newframe;
wire newpulse,one,two,three,four,five,six,seven,up;

DS3 d1 (clk,reset,count,data,sel,in,pulse,subframe,f0,f1,F0,F1,all,newframe,newpulse,one,two,three,four,five,six,seven,up);
initial
begin
  clk=1;
  reset=1;
  #10 reset=0;
end

always #10 clk=~clk;
initial
#1000000 $finish;

endmodule
```

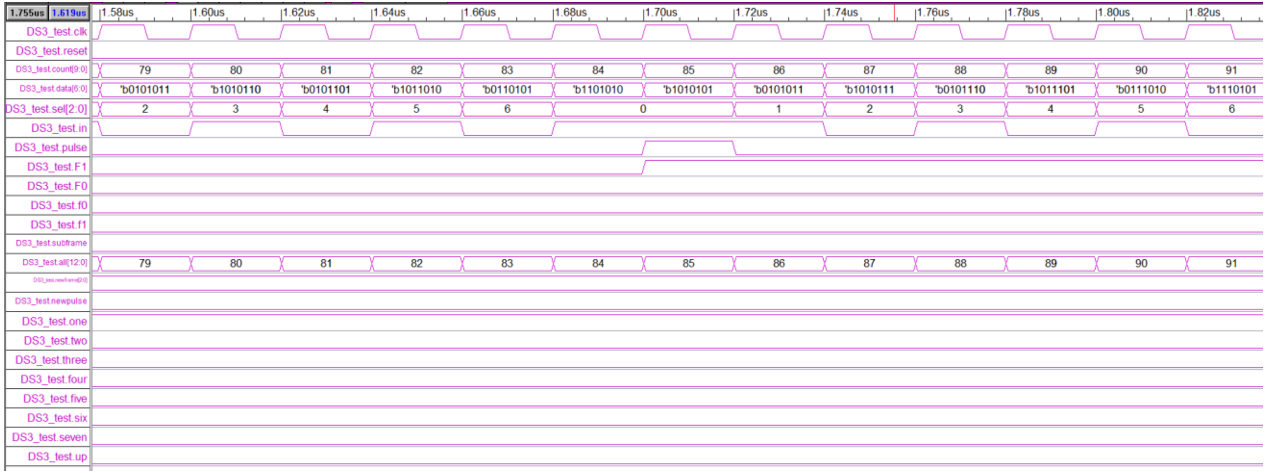
本次執行的檔案已以往的測試檔方式太龐大跑不了且過於不便，所以利用選擇線方式進行傳輸，主要程式皆在主體檔

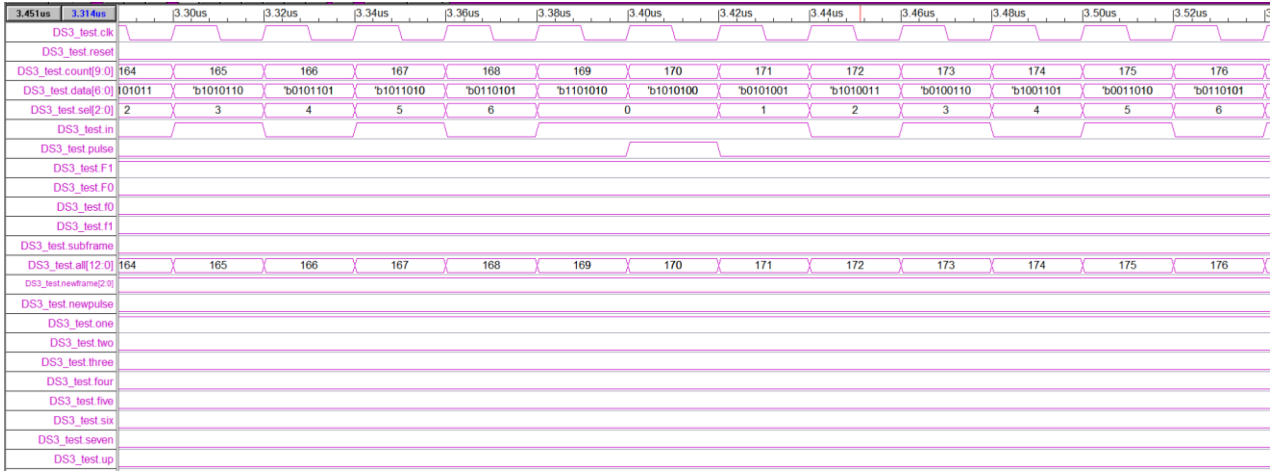
完整視圖

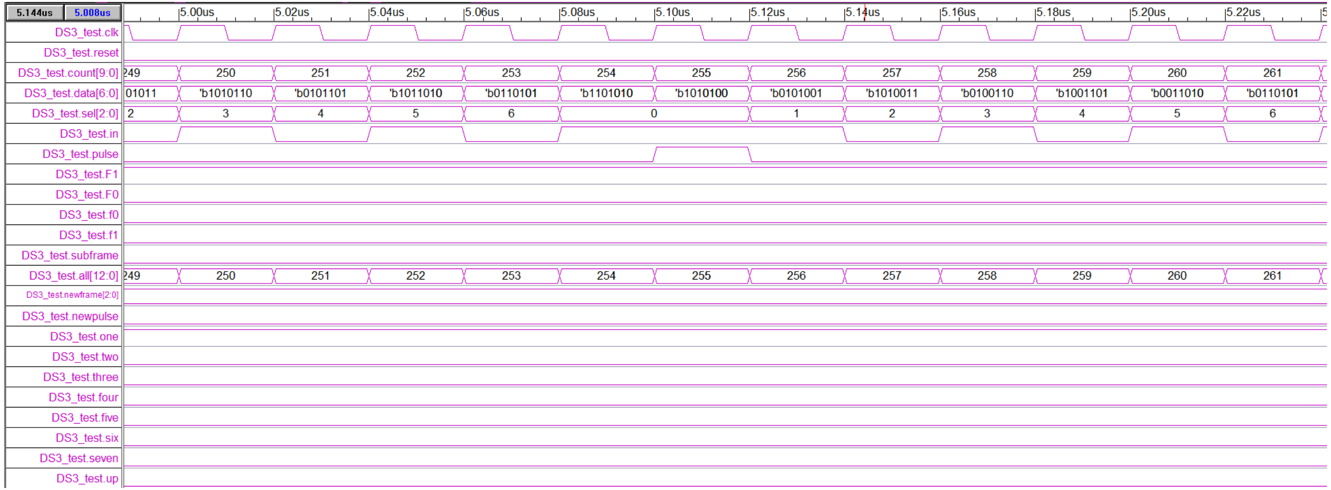

448.3us
380.3us
0us
50us
100us
150us
200us
250us
300us
350us
400us
450us
500us
550us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
DS3_test.data[6:0]
DS3_test.sel
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
DS3_test.newframe[2:0]
'b000 'b001 'b011 'b110 'b101 'b010 'b101 'b011 'b111 'b110 'b101 'b010 'b101 'b011 'b111 'b110 'b101 'b010 'b101 'b011 'b111 'b110 'b101 'b010 'b101 'b011 'b111 'b110 'b101 'b010 'b101 'b011 'b111 'b110 'b101 'b010 'b1
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

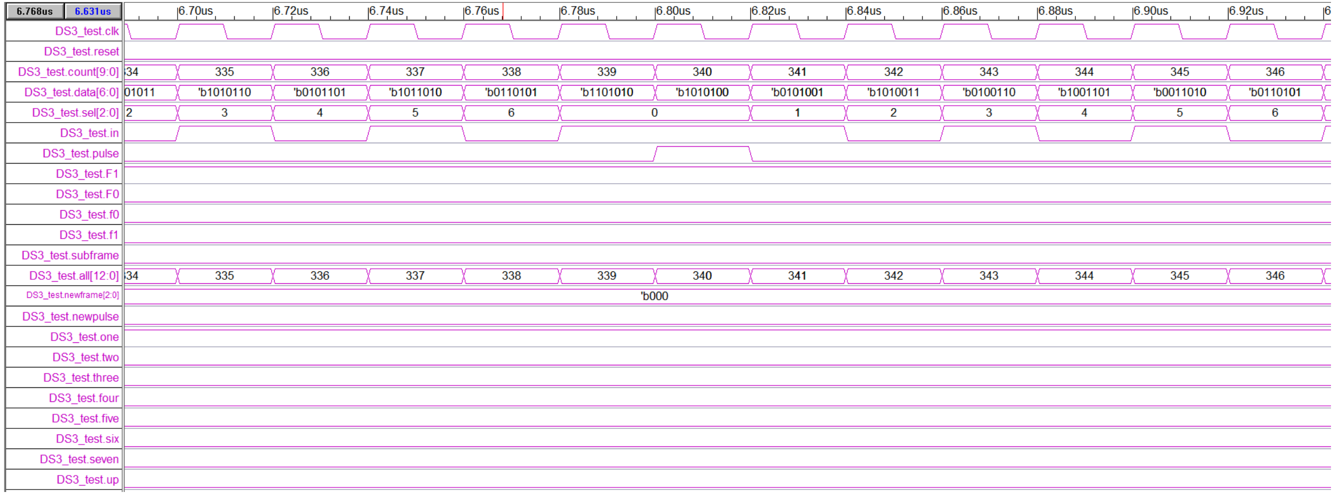
細部視圖

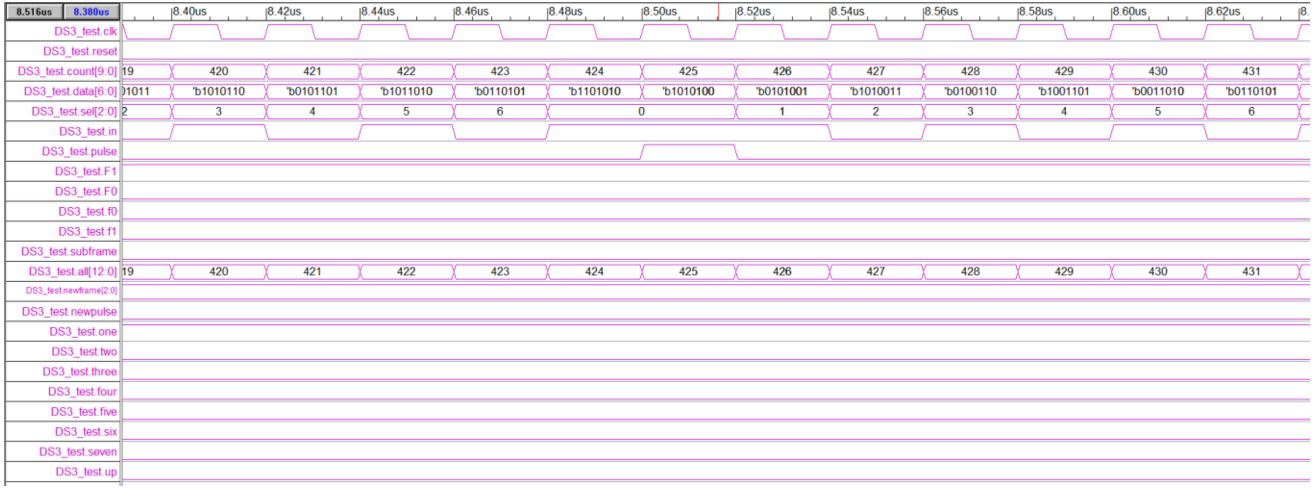

131.0ns
-6.000ns
0ns 10ns 20ns 30ns 40ns 50ns 60ns 70ns 80ns 90ns 100ns 110ns 120ns 130ns 140ns 150ns 160ns 170ns 180ns 190ns 200ns 210ns 220ns 230ns 240ns 250ns
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0] 0 1 2 3 4 5 6 7 8 9 10 11 12
DS3_test.data[6:0] 'b0000001 'b0000011 'b0000111 'b0001110 'b0011101 'b0111010 'b1110101 'b1101010 'b1010101 'b0101011 'b1010110 'b0101101 'b1011010
DS3_test.sel[2:0] 0 1 2 3 4 5 6 0 1 2 3 4 5
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0] 0 1 2 3 4 5 6 7 8 9 10 11 12
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

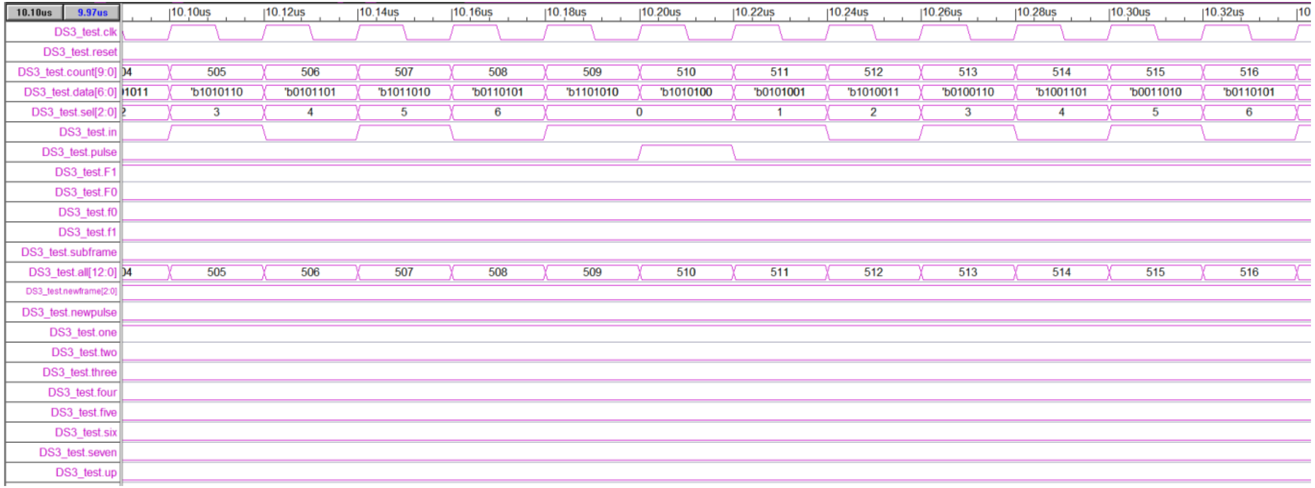

1.755us
1.619us
1.58us
1.60us
1.62us
1.64us
1.66us
1.68us
1.70us
1.72us
1.74us
1.76us
1.78us
1.80us
1.82us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
79 80 81 82 83 84 85 86 87 88 89 90 91
DS3_test.data[6:0]
'b0101011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010101 'b0101011 'b1010111 'b0101110 'b1011101 'b0111010 'b1110101
DS3_test.sel[2:0]
2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
79 80 81 82 83 84 85 86 87 88 89 90 91
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

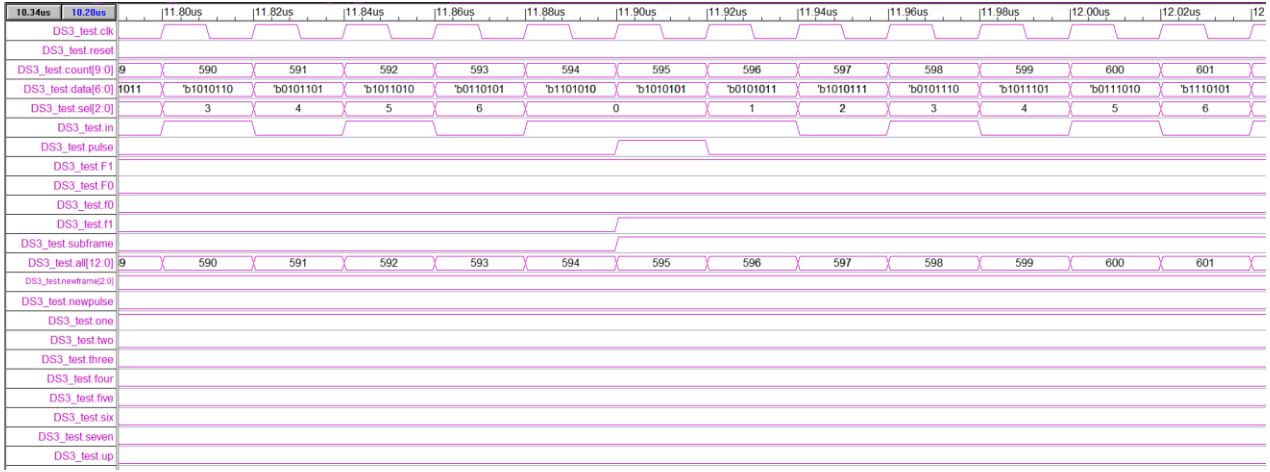

3.451us
3.314us
3.30us
3.32us
3.34us
3.36us
3.38us
3.40us
3.42us
3.44us
3.46us
3.48us
3.50us
3.52us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
164 165 166 167 168 169 170 171 172 173 174 175 176
DS3_test.data[6:0]
101011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010100 'b0101001 'b1010011 'b0100110 'b1001101 'b0011010 'b0110101
DS3_test.sel[2:0]
2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
164 165 166 167 168 169 170 171 172 173 174 175 176
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

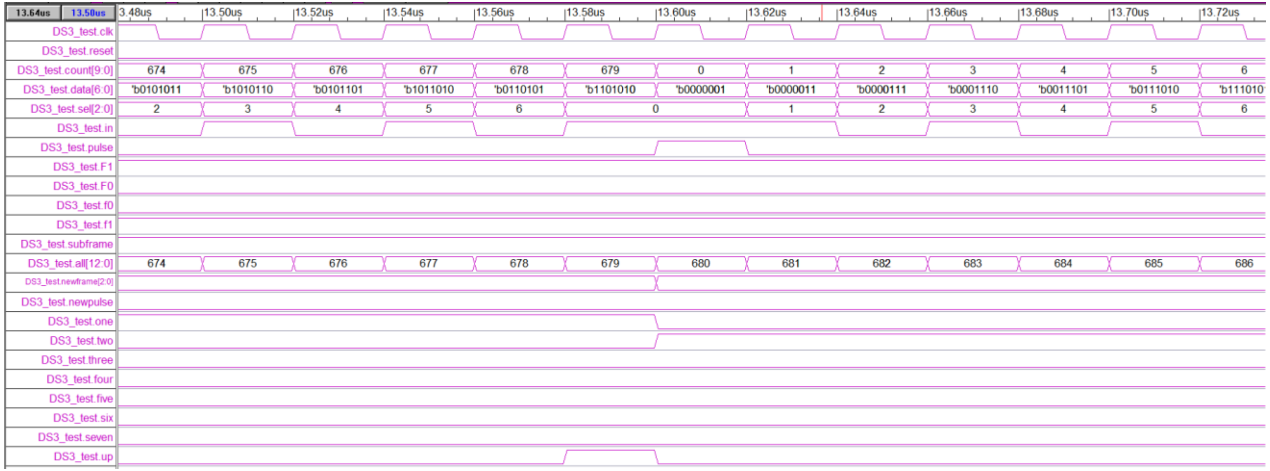

5.144us
5.008us
5.00us
5.02us
5.04us
5.06us
5.08us
5.10us
5.12us
5.14us
5.16us
5.18us
5.20us
5.22us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
249 250 251 252 253 254 255 256 257 258 259 260 261
DS3_test.data[6:0]
01011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010100 'b0101001 'b1010011 'b0100110 'b1001101 'b0011010 'b0110101
DS3_test.sel[2:0]
2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
249 250 251 252 253 254 255 256 257 258 259 260 261
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

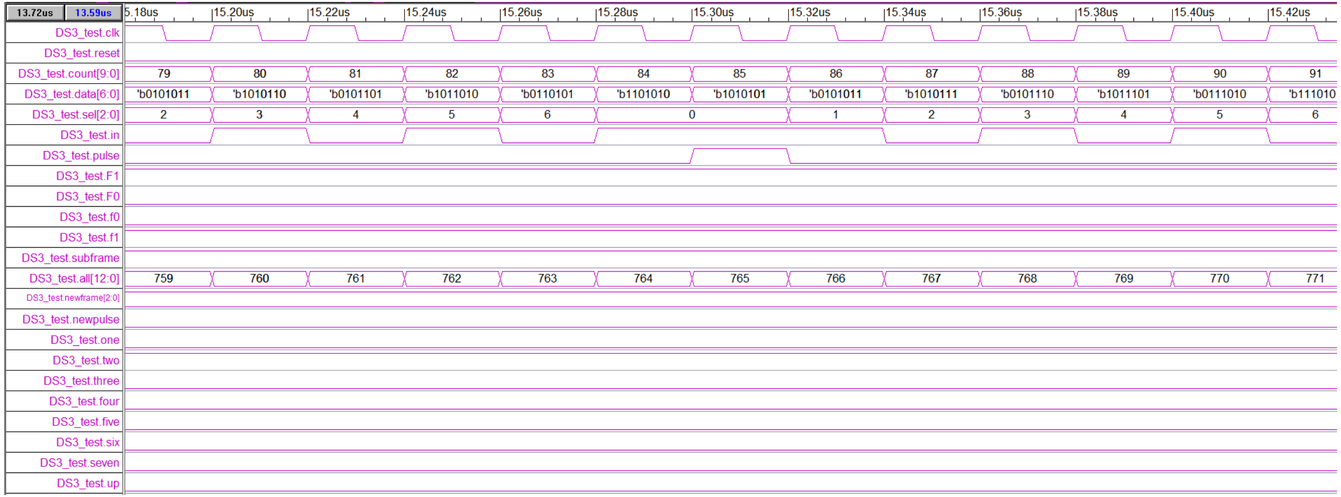

6.768us
6.631us
6.70us
6.72us
6.74us
6.76us
6.78us
6.80us
6.82us
6.84us
6.86us
6.88us
6.90us
6.92us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
34 335 336 337 338 339 340 341 342 343 344 345 346
DS3_test.data[6:0]
01011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010100 'b0101001 'b1010011 'b0100110 'b1001101 'b0011010 'b0110101
DS3_test.sel[2:0]
2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
34 335 336 337 338 339 340 341 342 343 344 345 346
DS3_test.newframe[2:0]
'b000
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

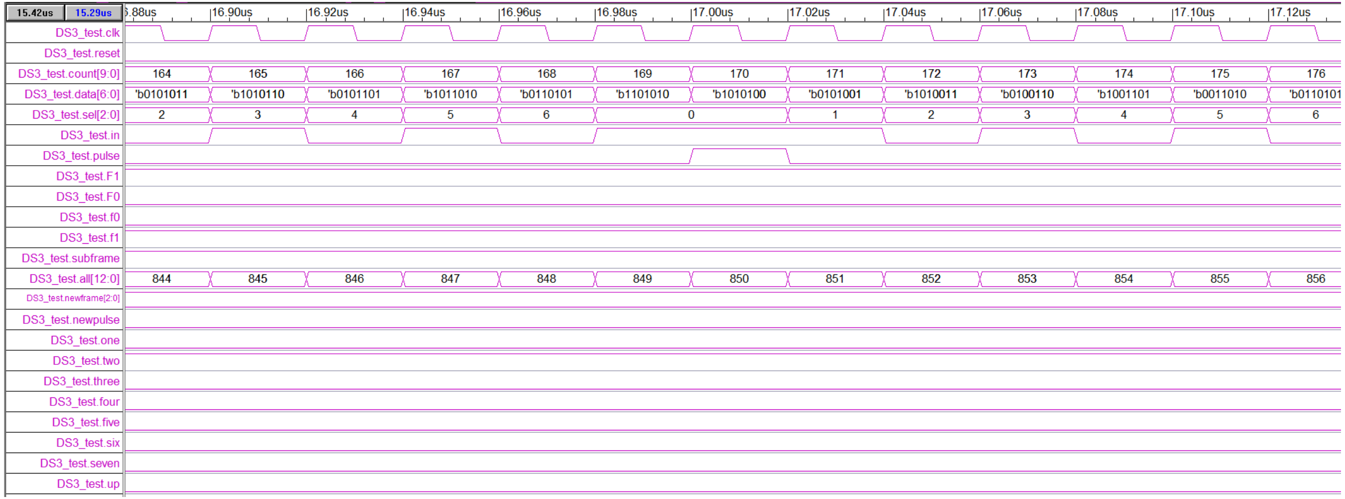

8.516us
8.380us
8.40us 8.42us 8.44us 8.46us 8.48us 8.50us 8.52us 8.54us 8.56us 8.58us 8.60us 8.62us 8.
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0] 19 420 421 422 423 424 425 426 427 428 429 430 431
DS3_test.data[6:0])1011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010100 'b0101001 'b1010011 'b0100110 'b1001101 'b0011010 'b0110101
DS3_test.sel[2:0] 2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0] 19 420 421 422 423 424 425 426 427 428 429 430 431
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

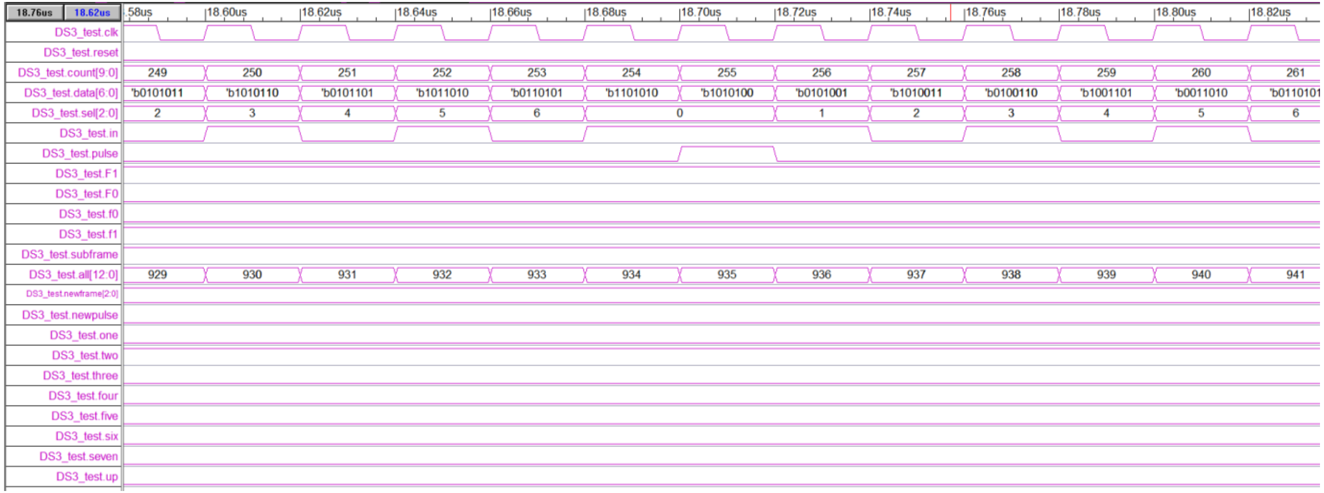

10.10us
9.97us
10.10us 10.12us 10.14us 10.16us 10.18us 10.20us 10.22us 10.24us 10.26us 10.28us 10.30us 10.32us 10
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0])4 505 506 507 508 509 510 511 512 513 514 515 516
DS3_test.data[6:0])1011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010100 'b0101001 'b1010011 'b0100110 'b1001101 'b0011010 'b0110101
DS3_test.sel[2:0] 2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0])4 505 506 507 508 509 510 511 512 513 514 515 516
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

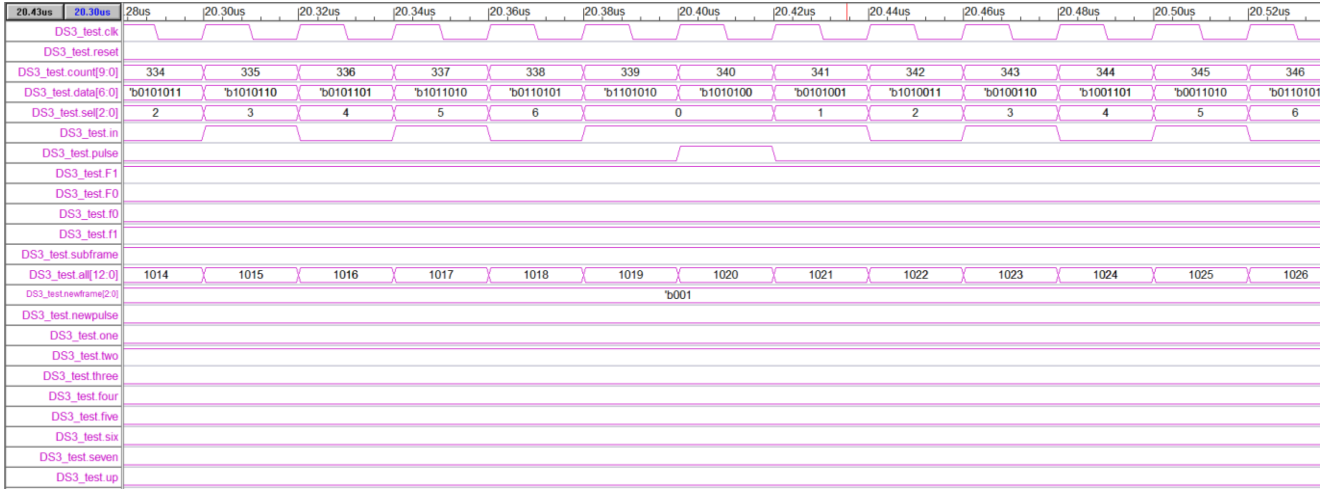

10.34us
10.20us
11.80us
11.82us
11.84us
11.86us
11.88us
11.90us
11.92us
11.94us
11.96us
11.98us
12.00us
12.02us
12
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
9
590
591
592
593
594
595
596
597
598
599
600
601
DS3_test.data[6:0]
1011
'b1010110
'b0101101
'b1011010
'b0110101
'b1101010
'b1010101
'b0101011
'b1010111
'b0101110
'b1011101
'b0111010
'b1110101
DS3_test.sel[2:0]
3
4
5
6
0
1
2
3
4
5
6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
9
590
591
592
593
594
595
596
597
598
599
600
601
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

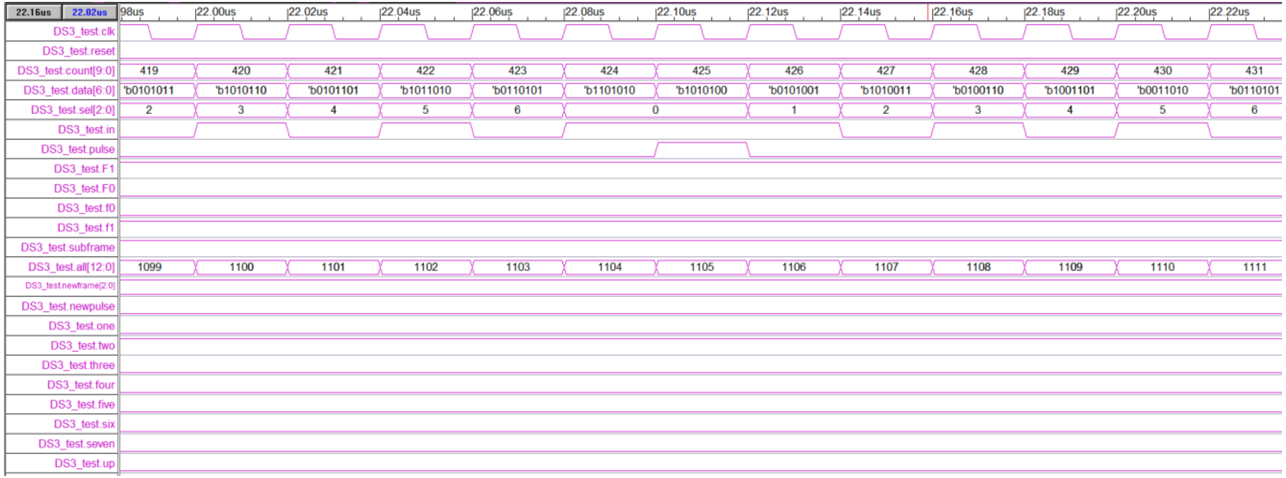

13.64us
13.50us
3.48us
13.50us
13.52us
13.54us
13.56us
13.58us
13.60us
13.62us
13.64us
13.66us
13.68us
13.70us
13.72us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
674
675
676
677
678
679
0
1
2
3
4
5
6
DS3_test.data[6:0]
'b0101011
'b1010110
'b0101101
'b1011010
'b0110101
'b1101010
'b0000001
'b0000011
'b0000111
'b0001110
'b0011101
'b0111010
'b111010
DS3_test.sel[2:0]
2
3
4
5
6
0
1
2
3
4
5
6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
674
675
676
677
678
679
680
681
682
683
684
685
686
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

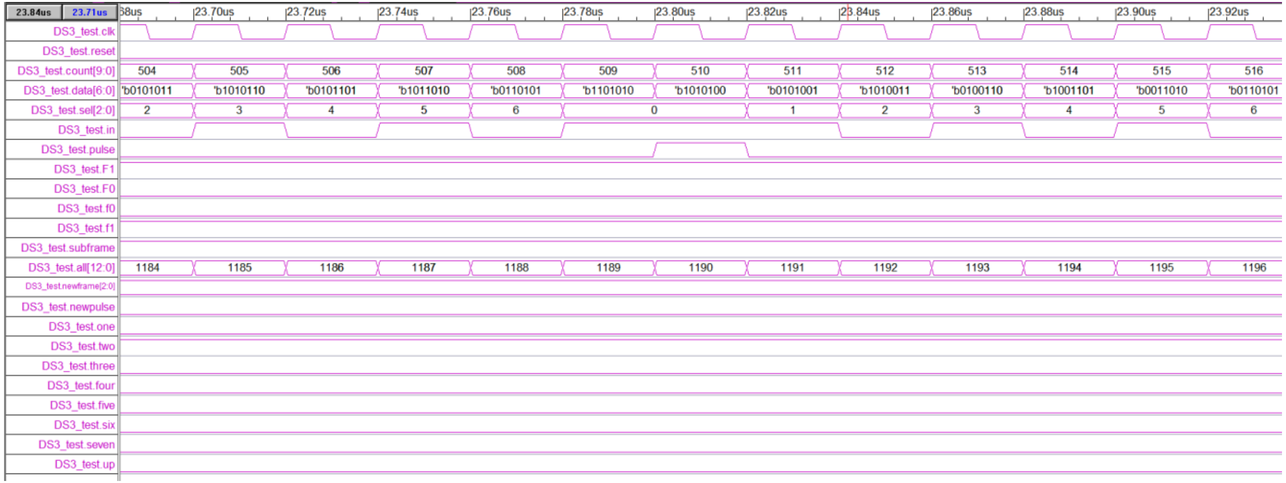

13.72us
13.59us
5.18us
15.20us
15.22us
15.24us
15.26us
15.28us
15.30us
15.32us
15.34us
15.36us
15.38us
15.40us
15.42us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
79
80
81
82
83
84
85
86
87
88
89
90
91
DS3_test.data[6:0]
'b0101011
'b1010110
'b0101101
'b1011010
'b0110101
'b1101010
'b1010101
'b0101011
'b1010111
'b0101110
'b1011101
'b0111010
'b111010
DS3_test.sel[2:0]
2
3
4
5
6
0
1
2
3
4
5
6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
759
760
761
762
763
764
765
766
767
768
769
770
771
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

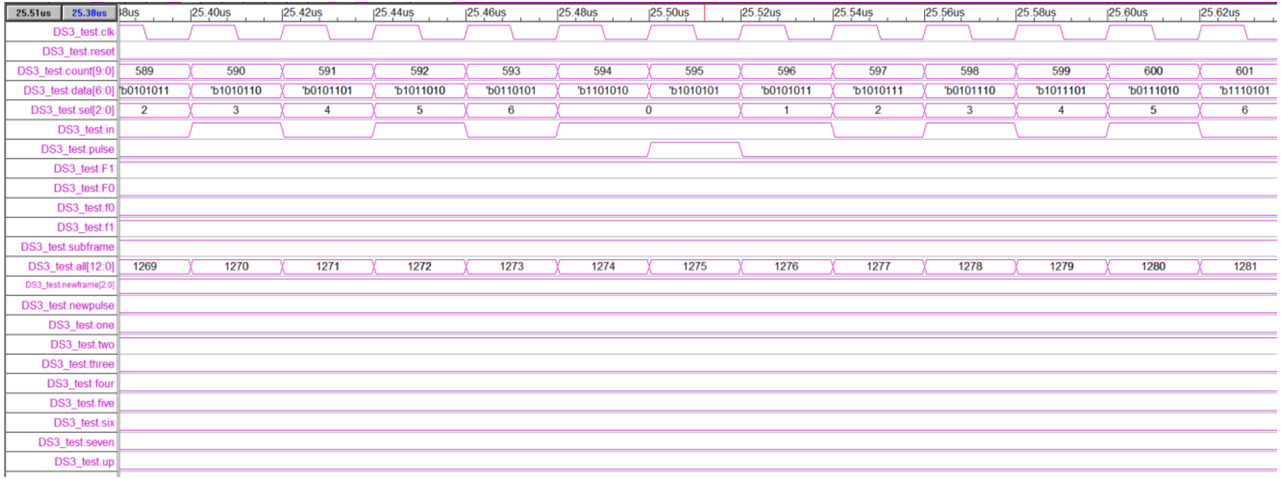

15.42us
15.29us
5.88us
16.90us
16.92us
16.94us
16.96us
16.98us
17.00us
17.02us
17.04us
17.06us
17.08us
17.10us
17.12us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
164
165
166
167
168
169
170
171
172
173
174
175
176
DS3_test.data[6:0]
'b0101011
'b1010110
'b0101101
'b1011010
'b0110101
'b1101010
'b1010100
'b0101001
'b1010011
'b0100110
'b1001101
'b0011010
'b0110101
DS3_test.sel[2:0]
2
3
4
5
6
0
1
2
3
4
5
6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
844
845
846
847
848
849
850
851
852
853
854
855
856
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

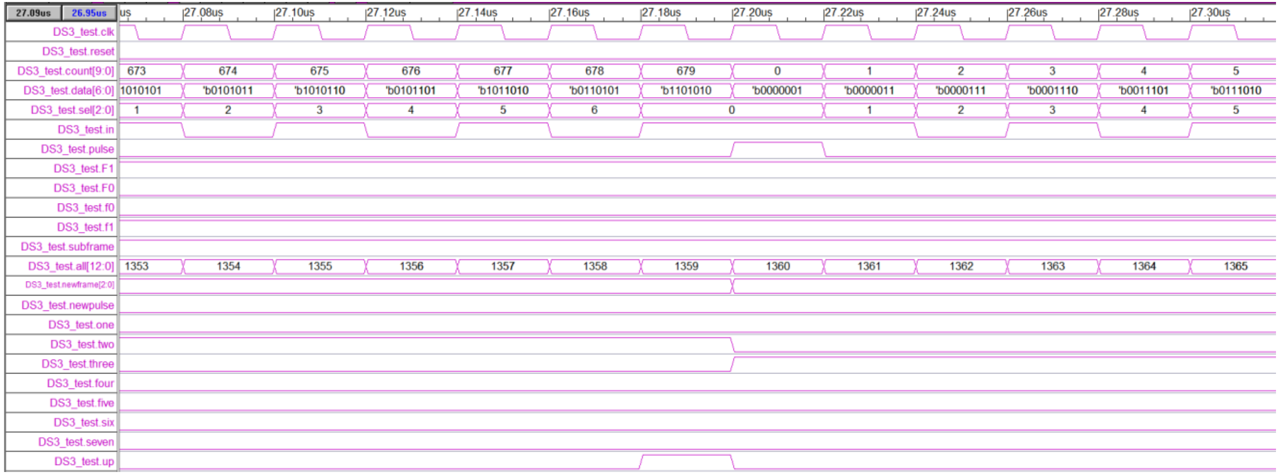

18.76us
18.62us
58us
18.60us
18.62us
18.64us
18.66us
18.68us
18.70us
18.72us
18.74us
18.76us
18.78us
18.80us
18.82us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
249 250 251 252 253 254 255 256 257 258 259 260 261
DS3_test.data[6:0]
'b0101011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010100 'b0101001 'b1010011 'b0100110 'b1001101 'b0011010 'b0110101
DS3_test.sel[2:0]
2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
929 930 931 932 933 934 935 936 937 938 939 940 941
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up


20.43us
20.30us
28us
20.30us
20.32us
20.34us
20.36us
20.38us
20.40us
20.42us
20.44us
20.46us
20.48us
20.50us
20.52us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
334 335 336 337 338 339 340 341 342 343 344 345 346
DS3_test.data[6:0]
'b0101011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010100 'b0101001 'b1010011 'b0100110 'b1001101 'b0011010 'b0110101
DS3_test.sel[2:0]
2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026
DS3_test.newframe[2:0]
'b001
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up


22.16us
22.02us
98us
22.00us
22.02us
22.04us
22.06us
22.08us
22.10us
22.12us
22.14us
22.16us
22.18us
22.20us
22.22us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
419 420 421 422 423 424 425 426 427 428 429 430 431
DS3_test.data[6:0]
'b0101011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010100 'b0101001 'b1010011 'b0100110 'b1001101 'b0011010 'b0110101
DS3_test.sel[2:0]
2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up


23.84us
23.71us
68us
23.70us
23.72us
23.74us
23.76us
23.78us
23.80us
23.82us
23.84us
23.86us
23.88us
23.90us
23.92us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
504 505 506 507 508 509 510 511 512 513 514 515 516
DS3_test.data[6:0]
'b0101011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010100 'b0101001 'b1010011 'b0100110 'b1001101 'b0011010 'b0110101
DS3_test.sel[2:0]
2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0]
1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up


25.51us
25.38us
8us 25.40us 25.42us 25.44us 25.46us 25.48us 25.50us 25.52us 25.54us 25.56us 25.58us 25.60us 25.62us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0] 589 590 591 592 593 594 595 596 597 598 599 600 601
DS3_test.data[6:0] 'b0101011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b1010101 'b0101011 'b1010111 'b0101110 'b1011101 'b0111010 'b1110101
DS3_test.sel[2:0] 2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0] 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up


27.09us
26.95us
us 27.08us 27.10us 27.12us 27.14us 27.16us 27.18us 27.20us 27.22us 27.24us 27.26us 27.28us 27.30us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0] 673 674 675 676 677 678 679 0 1 2 3 4 5
DS3_test.data[6:0] 1010101 'b0101011 'b1010110 'b0101101 'b1011010 'b0110101 'b1101010 'b0000001 'b0000011 'b0000111 'b0001110 'b0011101 'b0111010
DS3_test.sel[2:0] 1 2 3 4 5 6 0 1 2 3 4 5
DS3_test.in
DS3_test.pulse
DS3_test.F1
DS3_test.F0
DS3_test.f0
DS3_test.f1
DS3_test.subframe
DS3_test.all[12:0] 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365
DS3_test.newframe[2:0]
DS3_test.newpulse
DS3_test.one
DS3_test.two
DS3_test.three
DS3_test.four
DS3_test.five
DS3_test.six
DS3_test.seven
DS3_test.up

TX

傳送端1111111

程式介紹 - 初值設定

```
Module  DS3
(clk,reset,count,in,sel,allcount,data,pulse,F,subframe,Positionframe,PositionPulse,trigger);

input reset,clk;
output [9:0] count;
output [15:0] allcount;
output [3:0] sel;
output in,pulse,subframe;
output [6:0] data;
output [3:0] F;
output [2:0] Positionframe;
output PositionPulse;
output trigger;
```

```
reg [9:0] count;
reg [15:0] allcount;
reg [3:0] sel,F;
reg in,pulse,subframe;
reg [6:0] data;
reg [2:0] Positionframe;
reg PositionPulse;
reg trigger;
```

程式介紹

```
//count
always@(posedge clk)
begin
  if (reset)
  count=0;
  else if (count==679)
  count=0;
  else
  count=count+1;
end
always@(posedge clk)
begin
  if (reset)
  allcount=0;
  else if (allcount==4759)
  allcount=0;
  else
  allcount=allcount+1;
end
```

- 這裡利用正緣 clk 作為觸發
- 將選擇線還有計數器初值設為 0
- 讓計數器數 0~679 共 680 bits
- 這邊讓它不斷的數 0 到 6 全部需要共有 4760 bits

```
// multiplexer upcounter
always@(posedge clk)
begin
 if (reset)
     sel=0;
 else if (sel==6)
     sel=0;
 else
     sel=sel+1;
end
```

- sel 用來控制 in ，輸入有7組，所以數 0 到 6
- 達到輸入資料1111111 ，7組資料，用7條控制線控制

```
always@(posedge clk)
begin
 if (reset)
 in=1;
 else if (sel==0) in=1;
 else if (sel==1) in=1;
 else if (sel==2) in=1;
 else if (sel==3) in=1;
 else if (sel==4) in=1;
 else if (sel==5) in=1;
 else if (sel==6) in=1;
end
```

- 因為傳送之資料皆為1，所以將所有資料的控制線設為1

```
always@(posedge clk)
begin
  if (reset)
  data=7'b0000000;
  else
  data[6]=data[5];
  data[5]=data[4];
  data[4]=data[3];
  data[3]=data[2];
  data[2]=data[1];
  data[1]=data[0];
  data[0]=in;
end
```

- data[ ]用來一個一個的導入 in 的值

```
//C=0 F0=0
always@(posedge clk)
begin
  if (count==170 || count==255 || count==340 || count== 425 || count==510 || allcount==2720 || allcount==4080)
  data=7'b1111110;
end
```

- 將 0~4759 中需要以零表示之位置進行歸零動作。

```
always@(posedge clk)
begin
 if (reset)
 pulse = 1;
 else if (count==85 || count==170 ||
count==255 || count == 340 || count
==425 || count == 510 || count == 595
|| count ==0)
 pulse = 1;
 else
 pulse = 0;
end
```

- 透過偵測一列中之識別表示字元令pulse跳起之動作表示已經傳送完成1列中訊框部分資料
- 識別字元(依序)

  EX. F1、C11、F0、C12、F0、C13、F1

```
//F0 and F1 1001
always@(clk)
begin
 if(reset  || allcount==0)
 F=0;
 else if (count==85)
 F[3]=data[0];
 else if (count==255)
 F[2]=data[0];
 else if (count==425)
 F[1]=data[0];
 else if (count==595)
 F[0]=data[0];
end
```

- 在各個Frame將data[ ]資料引入F[3:0]中

```
//if F0 and F1 =1001 subframe HIGH
always@(posedge clk)
begin
  if (reset)
       subframe=0;
 if (F==4'b1001)
       subframe=1;
 if (allcount==0)
       subframe=0;
end
```

- 透過偵測data[0]送至F[3:0]之資料判定訊框1001時令subframe跳起之動作表示已經傳送完成7列資料

```
// X1 X2 P1 P2 M0 M1 M0 SISO
always@(posedge clk)
begin
  if(reset)
Positionframe=3'b001;
  else if (count==0)
begin
Positionframe[2]=Positionframe[1];
Positionframe[1]=Positionframe[0];
Positionframe[0]=data[0];
end
end
```

- 將data[0]以序進序出方式傳送至Positionframe來準備之後對齊符號之判定

```
// find 010
always@(posedge clk)
begin
  if (reset)
PositionPulse=0;
else if (Positionframe==3'b010)
PositionPulse=1;
else
PositionPulse=0;
end
```

- 透過判定M0(2720)、M1(3400)、M0(4080)分別為0、1、0來進行資料對齊識別，並使用一個位置識別波形表示。

```
// trigger
always@(posedge clk)
begin
  if(reset)
trigger=0;
else if (count==679)
trigger=1;
else
trigger=0;
end

endmodule
```

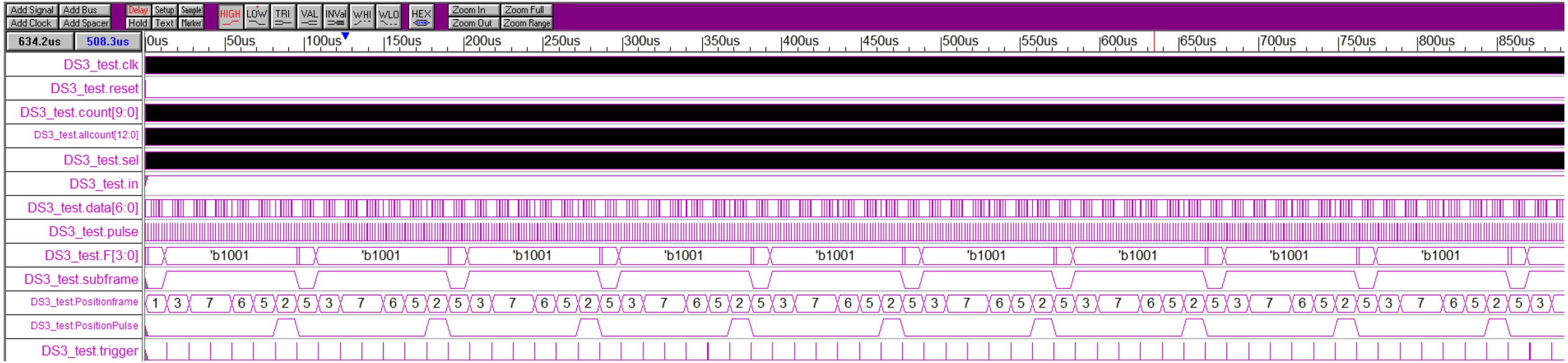
- 最後透過偵測一列中之識別表示字元令pulse跳起之動作表示已經傳送完成1列中整串資料。

測試檔

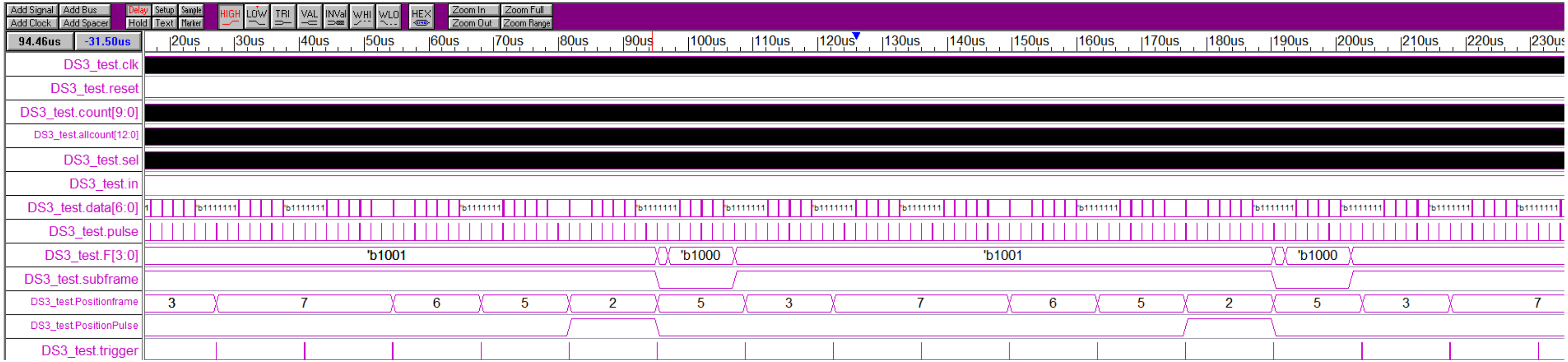
```
module DS3_test;
reg clk,reset;
wire [9:0] count;
wire [15:0] allcount;
wire [3:0] sel;
wire in,pulse,subframe;
wire [6:0] data;
wire [3:0] F;
wire [2:0] Positionframe;
wire PositionPulse;
wire trigger;

DS3 d1 (clk,reset,count,in,sel,allcount,data,pulse,F,subframe,Positionframe,PositionPulse,trigger);
initial
begin
  clk=1;
  reset=1;
  #10 reset=0;
end

always #10 clk=~clk;
initial
#1000000 $finish;

endmodule
```

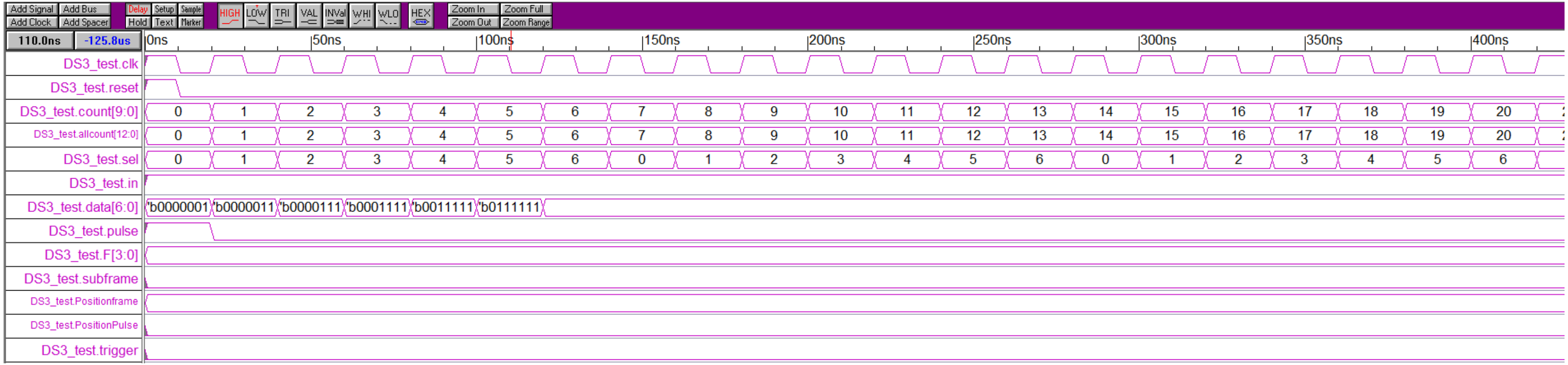
本次執行的檔案已以往的測試檔方式太龐大跑不了且過於不便，所以利用選擇線方式進行傳輸，主要程式皆在主體檔

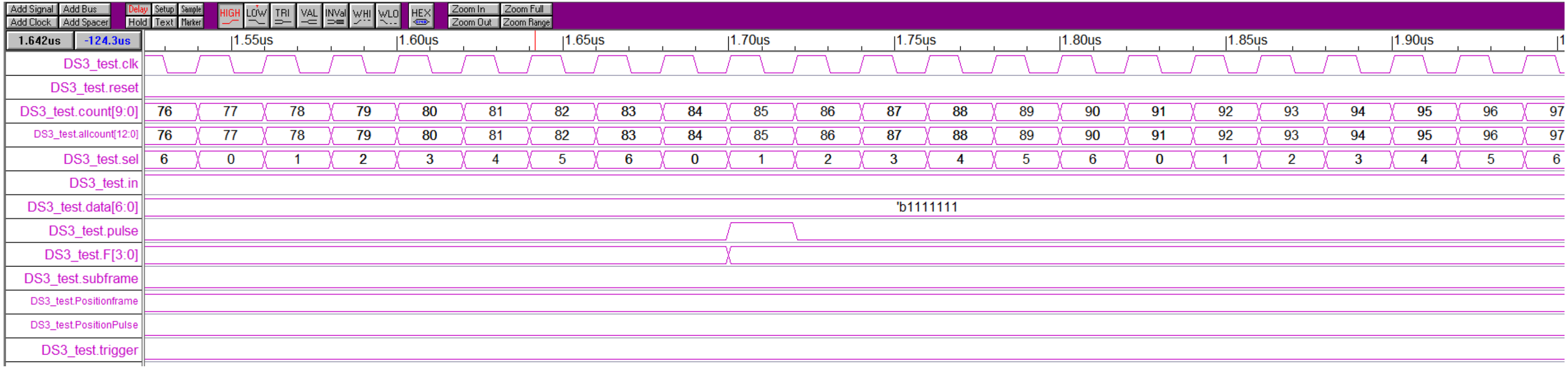
完整視圖

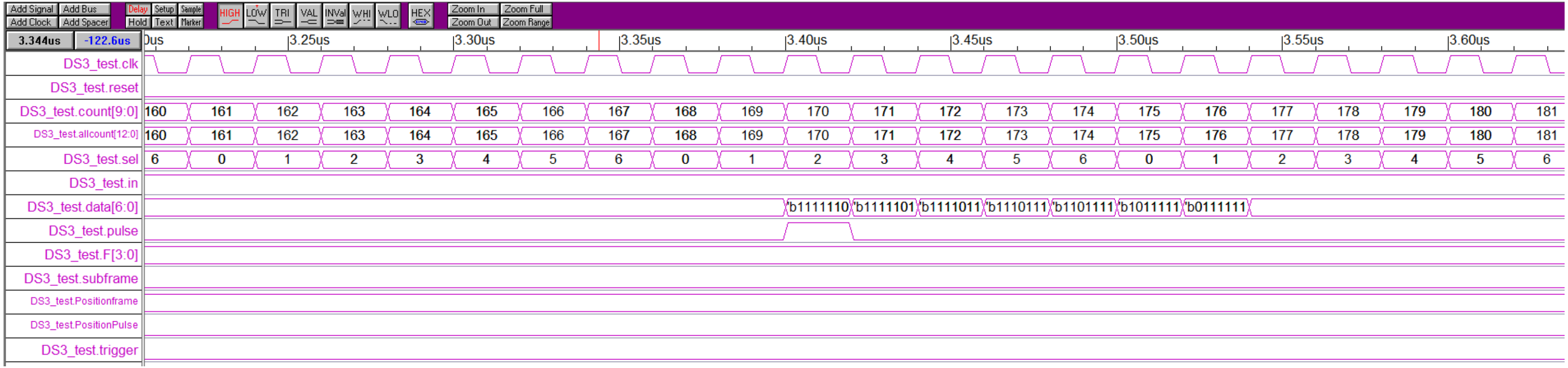

Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
634.2us
508.3us
0us
50us
100us
150us
200us
250us
300us
350us
400us
450us
500us
550us
600us
650us
700us
750us
800us
850us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
DS3_test.allcount[12:0]
DS3_test.sel
DS3_test.in
DS3_test.data[6:0]
DS3_test.pulse
DS3_test.F[3:0]
'b1001
DS3_test.subframe
DS3_test.Positionframe
1 3 7 6 5 2 5 3 7 6 5 2 5 3 7 6 5 2 5 3 7 6 5 2 5 3 7 6 5 2 5 3 7 6 5 2 5 3 7 6 5 2 5 3 7 6 5 2 5 3 7 6 5 2 5 3
DS3_test.PositionPulse
DS3_test.trigger

細部視圖


Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
94.46us
-31.50us
20us
30us
40us
50us
60us
70us
80us
90us
100us
110us
120us
130us
140us
150us
160us
170us
180us
190us
200us
210us
220us
230us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
DS3_test.allcount[12:0]
DS3_test.sel
DS3_test.in
DS3_test.data[6:0]
'b1111111
DS3_test.pulse
DS3_test.F[3:0]
'b1001
'b1000
'b1001
'b1000
DS3_test.subframe
DS3_test.Positionframe
3
7
6
5
2
5
3
7
6
5
2
5
3
7
DS3_test.PositionPulse
DS3_test.trigger


Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
110.0ns
-125.8us
0ns
50ns
100ns
150ns
200ns
250ns
300ns
350ns
400ns
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
DS3_test.allcount[12:0]
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
DS3_test.sel
0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.data[6:0]
'b0000001 'b0000011 'b0000111 'b0001111 'b0011111 'b0111111
DS3_test.pulse
DS3_test.F[3:0]
DS3_test.subframe
DS3_test.Positionframe
DS3_test.PositionPulse
DS3_test.trigger


Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
1.642us
-124.3us
1.55us
1.60us
1.65us
1.70us
1.75us
1.80us
1.85us
1.90us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
DS3_test.allcount[12:0]
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
DS3_test.sel
6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6
DS3_test.in
DS3_test.data[6:0]
'b1111111
DS3_test.pulse
DS3_test.F[3:0]
DS3_test.subframe
DS3_test.Positionframe
DS3_test.PositionPulse
DS3_test.trigger



3.344us -122.6us

DS3_test.clk

DS3_test.reset

DS3_test.count[9:0] 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181

DS3_test.allcount[12:0] 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181

DS3_test.sel 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6

DS3_test.in

DS3_test.data[6:0] 'b1111110 'b1111101 'b1111011 'b1110111 'b1101111 'b1011111 'b0111111

DS3_test.pulse

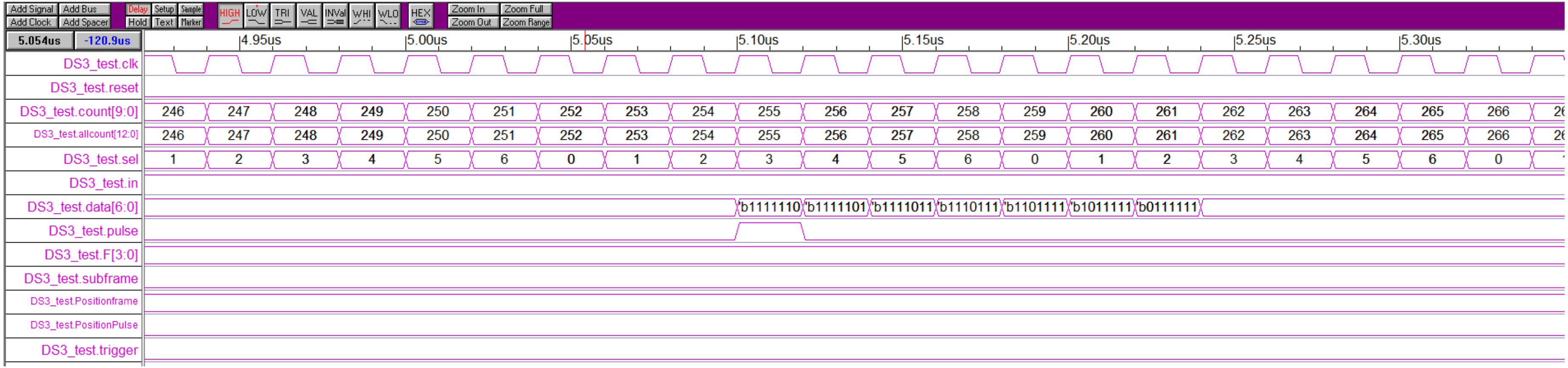
DS3_test.F[3:0]

DS3_test.subframe

DS3_test.Positionframe

DS3_test.PositionPulse

DS3_test.trigger



5.054us -120.9us

DS3_test.clk

DS3_test.reset

DS3_test.count[9:0] 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 26

DS3_test.allcount[12:0] 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 26

DS3_test.sel 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1

DS3_test.in

DS3_test.data[6:0] 'b1111110 'b1111101 'b1111011 'b1110111 'b1101111 'b1011111 'b0111111

DS3_test.pulse

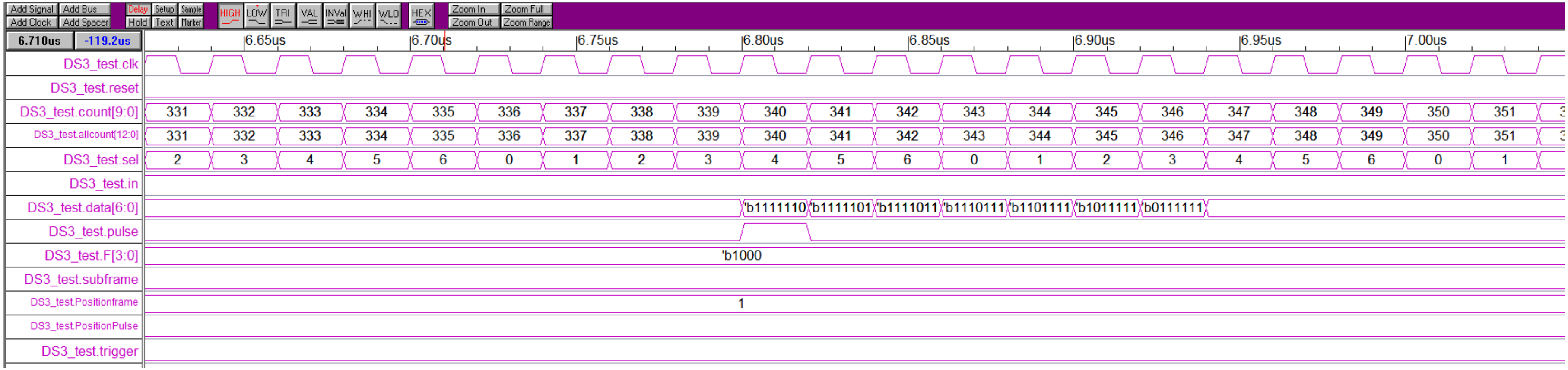
DS3_test.F[3:0]

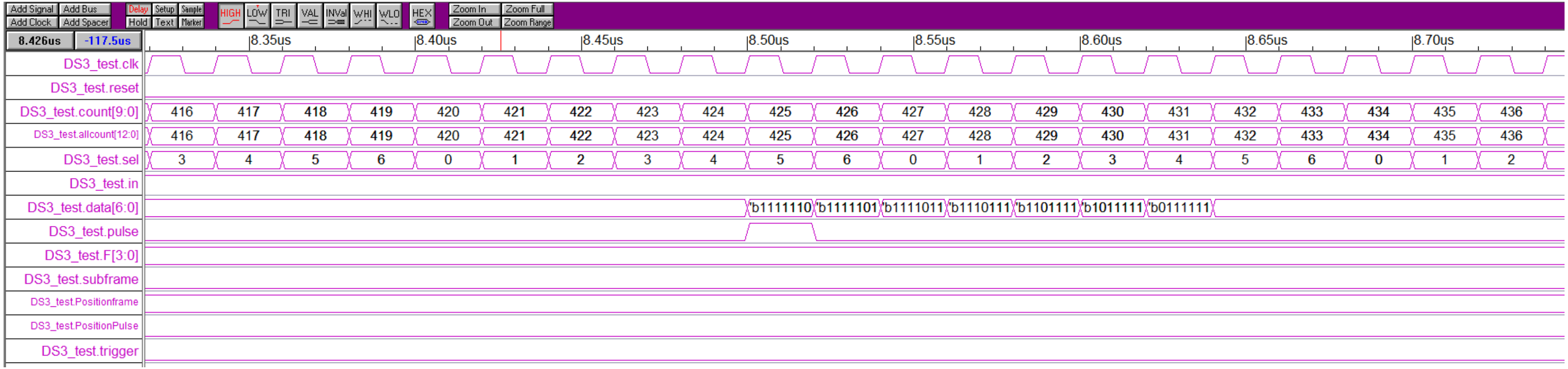
DS3_test.subframe

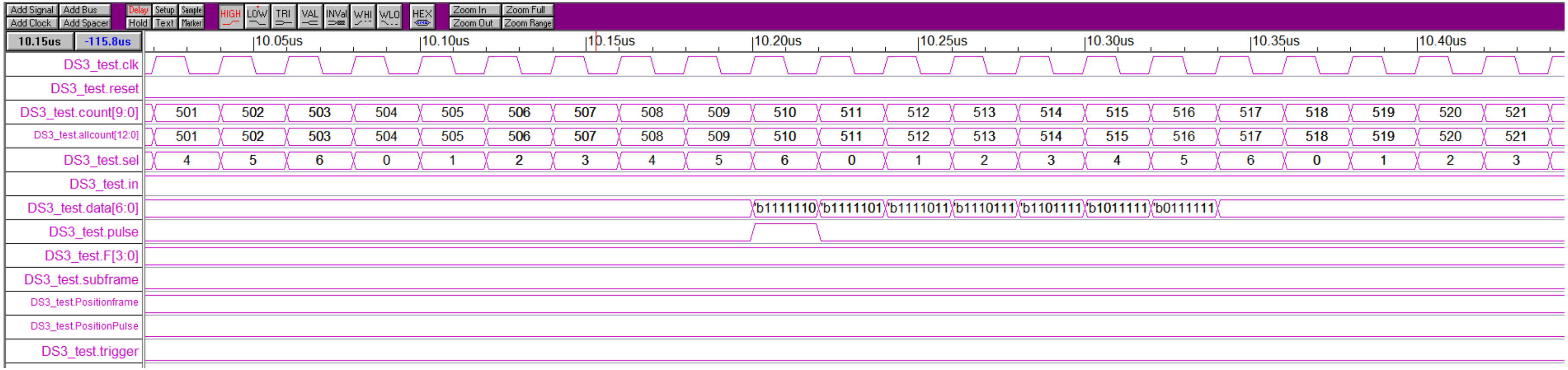
DS3_test.Positionframe

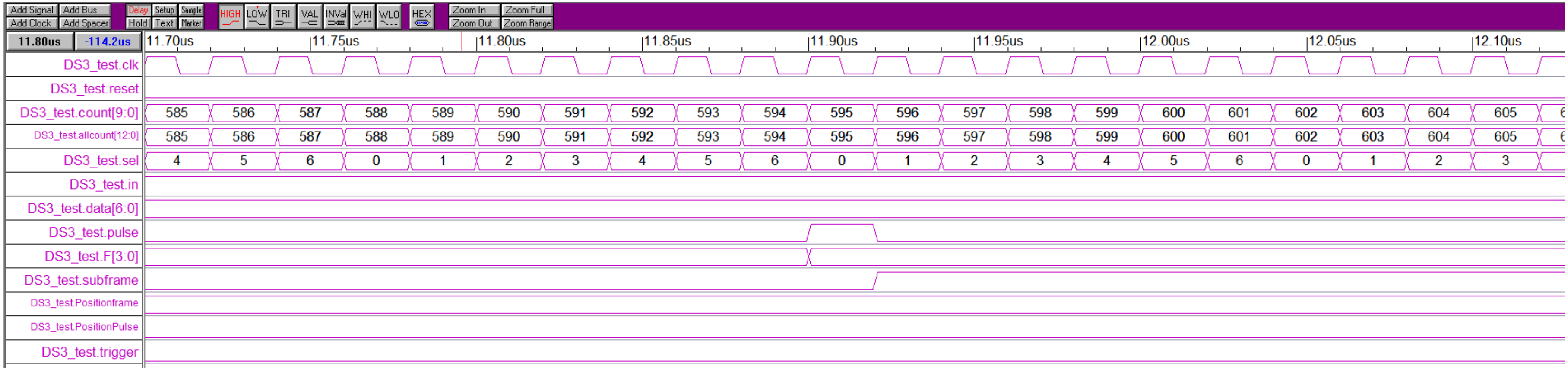
DS3_test.PositionPulse

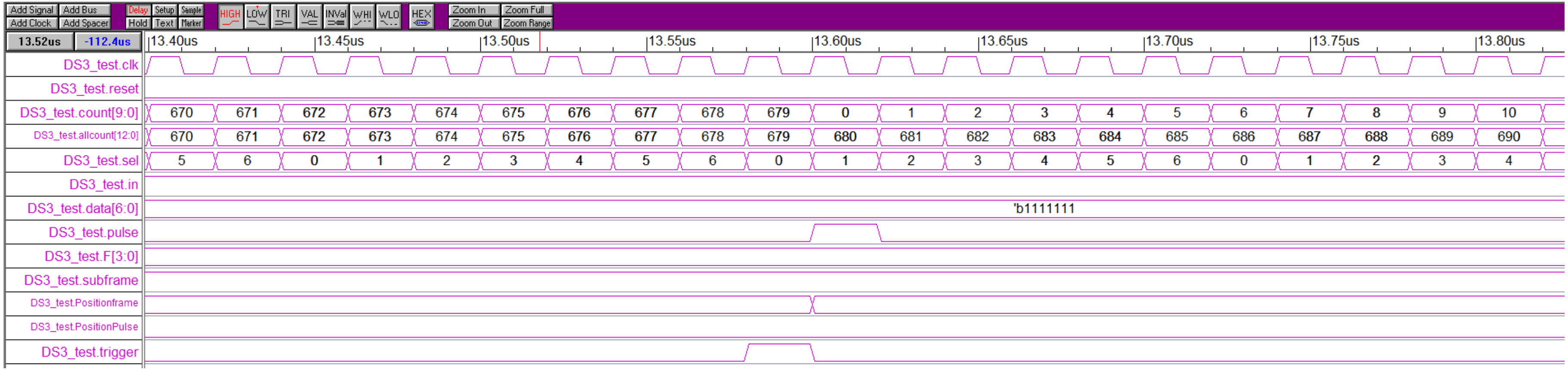
DS3_test.trigger

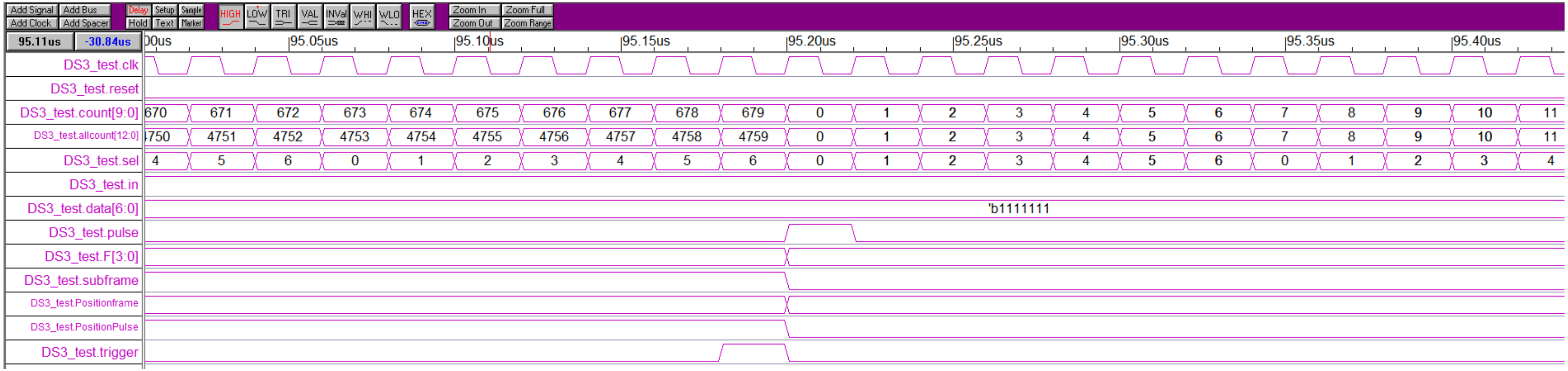

Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
6.710us
-119.2us
6.65us
6.70us
6.75us
6.80us
6.85us
6.90us
6.95us
7.00us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
DS3_test.allcount[12:0]
331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351
DS3_test.sel
2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1
DS3_test.in
DS3_test.data[6:0]
'b1111110 'b1111101 'b1111011 'b1110111 'b1101111 'b1011111 'b0111111
DS3_test.pulse
DS3_test.F[3:0]
'b1000
DS3_test.subframe
DS3_test.Positionframe
1
DS3_test.PositionPulse
DS3_test.trigger


Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
8.426us
-117.5us
8.35us
8.40us
8.45us
8.50us
8.55us
8.60us
8.65us
8.70us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436
DS3_test.allcount[12:0]
416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436
DS3_test.sel
3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2
DS3_test.in
DS3_test.data[6:0]
'b1111110 'b1111101 'b1111011 'b1110111 'b1101111 'b1011111 'b0111111
DS3_test.pulse
DS3_test.F[3:0]
DS3_test.subframe
DS3_test.Positionframe
DS3_test.PositionPulse
DS3_test.trigger


Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
10.15us
-115.8us
10.05us
10.10us
10.15us
10.20us
10.25us
10.30us
10.35us
10.40us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521
DS3_test.allcount[12:0]
501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521
DS3_test.sel
4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3
DS3_test.in
DS3_test.data[6:0]
'b1111110 'b1111101 'b1111011 'b1110111 'b1101111 'b1011111 'b0111111
DS3_test.pulse
DS3_test.F[3:0]
DS3_test.subframe
DS3_test.Positionframe
DS3_test.PositionPulse
DS3_test.trigger


Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
11.80us
-114.2us
11.70us
11.75us
11.80us
11.85us
11.90us
11.95us
12.00us
12.05us
12.10us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 6
DS3_test.allcount[12:0]
585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 6
DS3_test.sel
4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3
DS3_test.in
DS3_test.data[6:0]
DS3_test.pulse
DS3_test.F[3:0]
DS3_test.subframe
DS3_test.Positionframe
DS3_test.PositionPulse
DS3_test.trigger


Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
13.52us
-112.4us
13.40us
13.45us
13.50us
13.55us
13.60us
13.65us
13.70us
13.75us
13.80us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
670 671 672 673 674 675 676 677 678 679 0 1 2 3 4 5 6 7 8 9 10
DS3_test.allcount[12:0]
670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690
DS3_test.sel
5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4
DS3_test.in
DS3_test.data[6:0]
'b1111111
DS3_test.pulse
DS3_test.F[3:0]
DS3_test.subframe
DS3_test.Positionframe
DS3_test.PositionPulse
DS3_test.trigger


Add Signal
Add Bus
Add Clock
Add Spacer
Delay
Setup
Sample
Hold
Text
Marker
HIGH
LOW
TRI
VAL
INVal
WHI
WLO
HEX
Zoom In
Zoom Full
Zoom Out
Zoom Range
95.11us
-30.84us
00us
95.05us
95.10us
95.15us
95.20us
95.25us
95.30us
95.35us
95.40us
DS3_test.clk
DS3_test.reset
DS3_test.count[9:0]
670 671 672 673 674 675 676 677 678 679 0 1 2 3 4 5 6 7 8 9 10 11
DS3_test.allcount[12:0]
4750 4751 4752 4753 4754 4755 4756 4757 4758 4759 0 1 2 3 4 5 6 7 8 9 10 11
DS3_test.sel
4 5 6 0 1 2 3 4 5 6 0 1 2 3 4 5 6 0 1 2 3 4
DS3_test.in
DS3_test.data[6:0]
'b1111111
DS3_test.pulse
DS3_test.F[3:0]
DS3_test.subframe
DS3_test.Positionframe
DS3_test.PositionPulse
DS3_test.trigger

Thanks