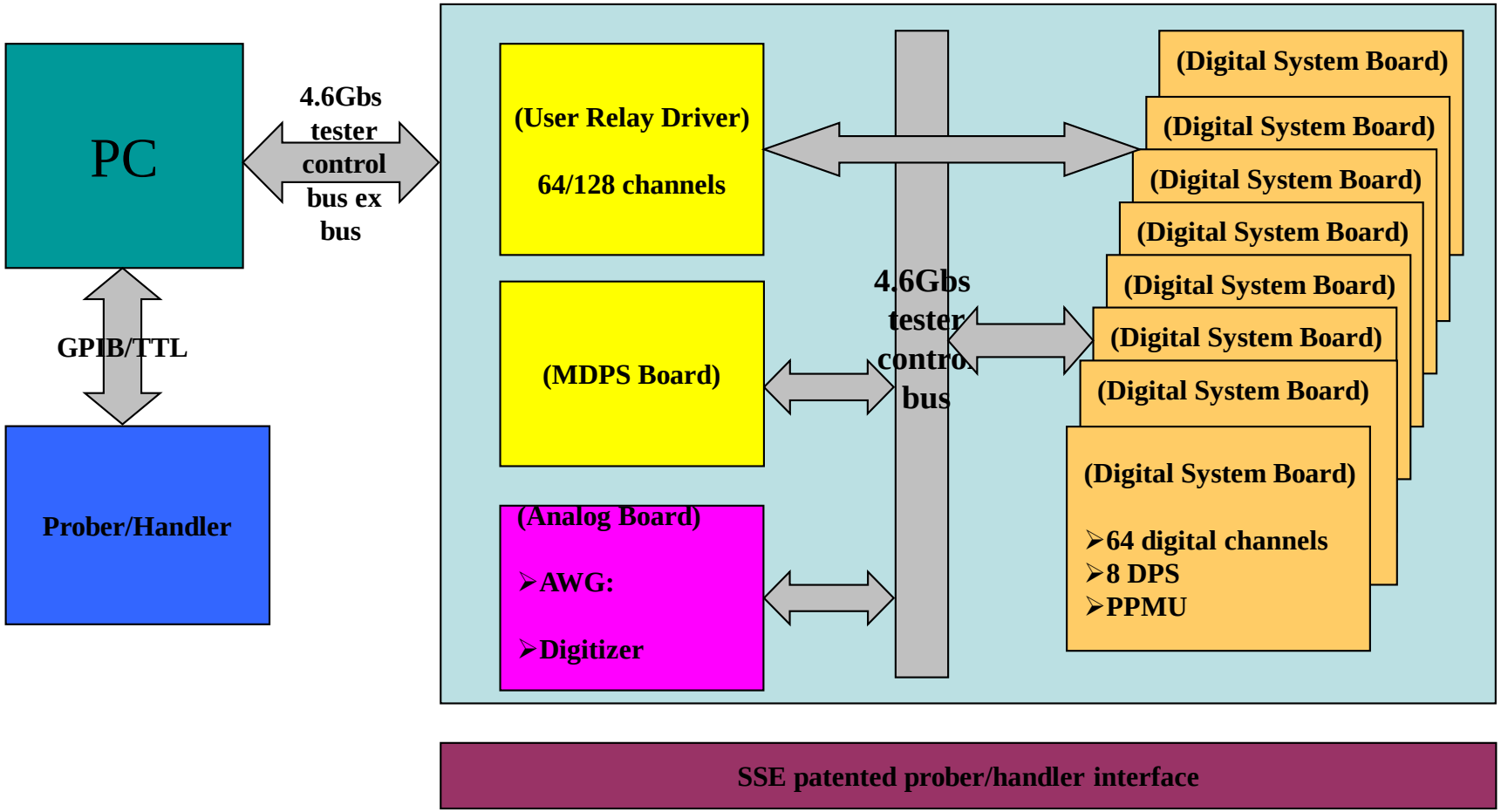
Sancode 100

Signality System Engineering Co., LTD.

Introduction

- 256/512/768 IO Channels with PPMU
- 100MHz test rate
- 32/64/96 DPS with gang function（future）
- Parallel test up to 32/64/96 DUTs
- 128/256M Pattern depth
- Per pin timing/format generator
- Timing/format change on the fly
- Per pin reference level
- Max. 125MHz clock rate
- Tester per board architecture
- Capable of multi-time domain patterns
- 4.6Gbs band width internal tester control bus
- Audio AWG/Digitizer option（future）

Sancode System Architecture



DPS - Specification

-8 DPS per digital system board
-DPS force change on the fly
-DPS high speed measure

- DPS

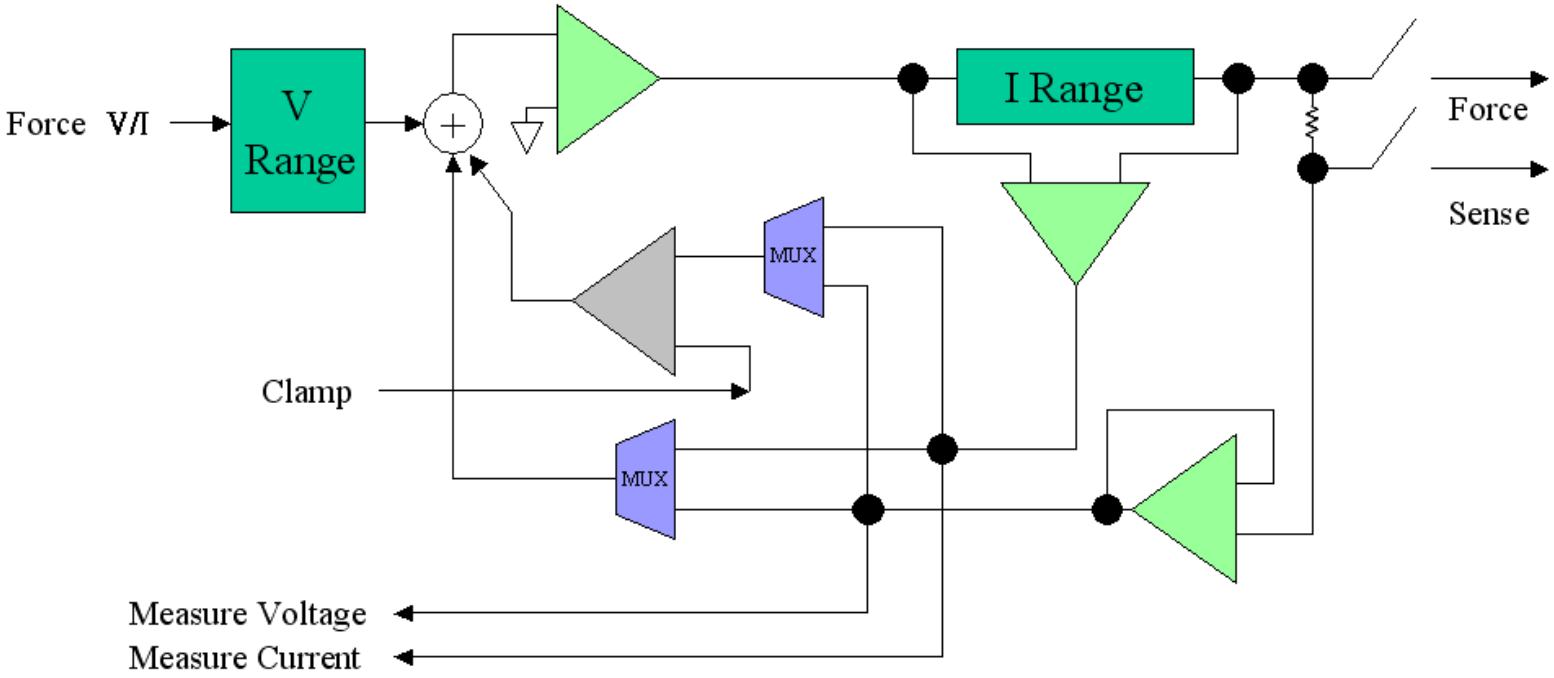
  Mode: FIMV, FVMI, FV, FI

  Voltage range (Resolution/Accuracy): +-2V(0.1mV/+-0.2%), +-4V(0.2mV/+-0.2%), +-8V(0.4mV/+-0.2%), -10V~+14V(0.5mV/+-0.2%)

  Current range (Resolution/Accuracy): 2uA(0.1na/+-0.7%), 20uA(1na/+-0.5%), 200uA(10na/+-0.5%), 2mA(100na/+-0.5%), 20mA(1ua/+-0.5%), 200mA(10ua/+-0.5%), 1A(100ua/+-0.5%)

  Programmable current limit: 10%~100% of Range

DPS/PMU Block Diagram



DPS Programming

- **HS_DPS_FV(DPS_No,Force_Voltage_Value,VI_Range,Clamp_I,Sense_Relay,DPS_Relay, Delay_Time);**
- **HS_DPS_FI(DPS_No,Force_Current_Value,VI_Range,Clamp_V,Sense_Relay,DPS_Relay, Delay_Time);**
- **HS_DPS_MI(DPS_No,Current_Low_Limit,Current_High_Limit,Test_Flag, Delay_Time);**
- **HS_DPS_MV(DPS_No,Voltage_Low_Limit,Voltage_High_Limit,Test_Flag, Delay_Time);**

Pin Electronics - Specification

-Pin count: 256 pins maximum
-Per pin resource: Level set, timing set & format set
-Timing & format change on the fly

- Driver
  Output voltage range: -2V~+7V
  Output impedance: 50 ohm
  Minimum pulse width: 3ns
  Tf/Tr: 1.6V/ns
  Tri-state voltage terminator: -2V~+7V

- Comparator
  Input voltage range: -2V~+7V
  Leakage: <1uA
  Active Load: 20mA
  Vtt: -2V~+7V

Pin Electronics – Specification

- Timing generator
  Period range: 10ns~512ns, 2ns resolution
  Timing set count: 128 sets maximum @ Period = 10ns
  Driver: Multi-Edges drive, 4ns minimum pulse width, 2ns resolution
  Multi-IO switch, 4ns minimum switch width, 2ns resolution

  Comparator: Multi-Windows strobe, 2ns minimum window width

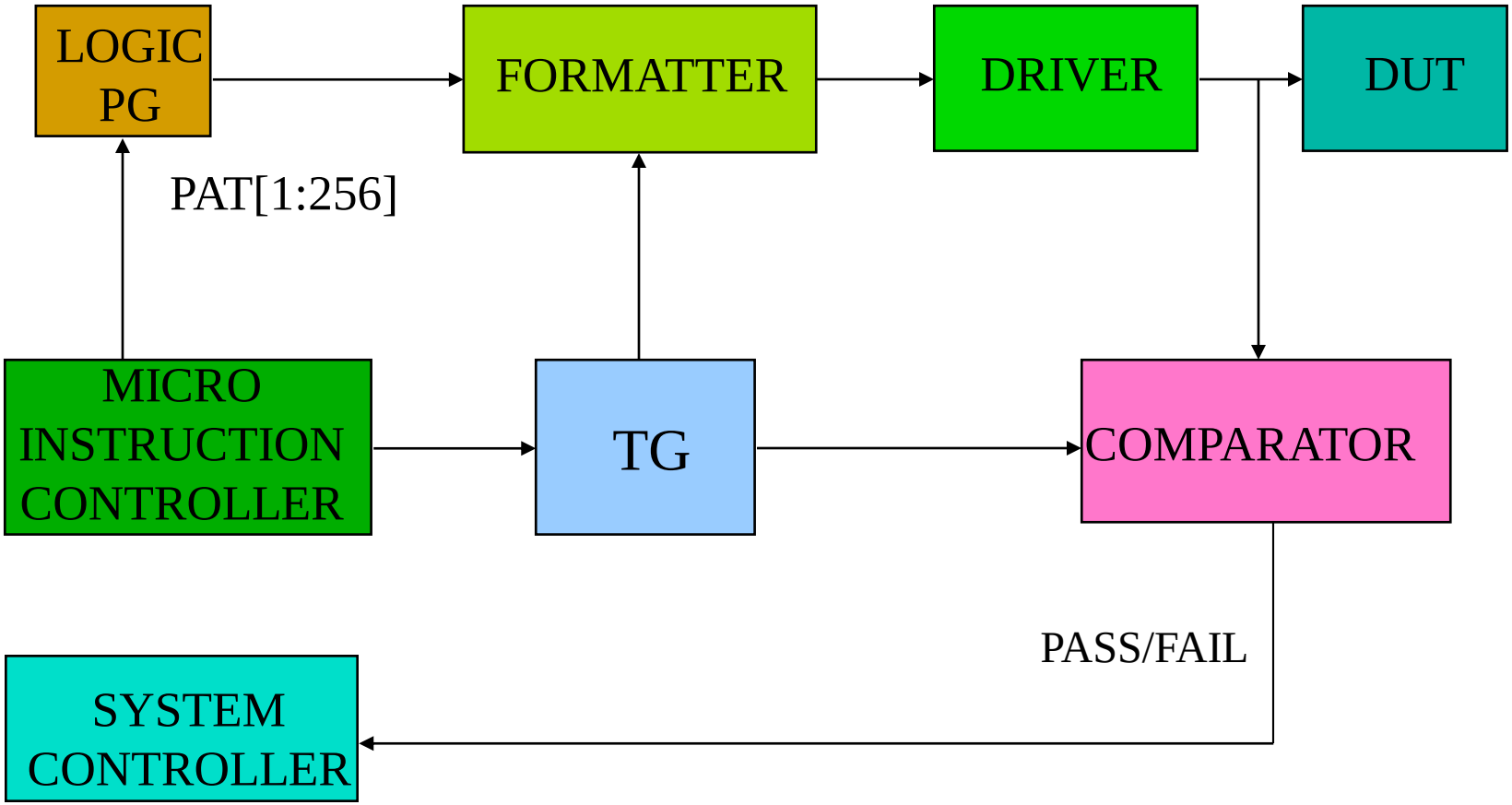
- Drive Formatter
  NRZ
  RZ
  RTO
  SBC

- IO Formatter
  NRZ
  RZ
  RTO
  SBC

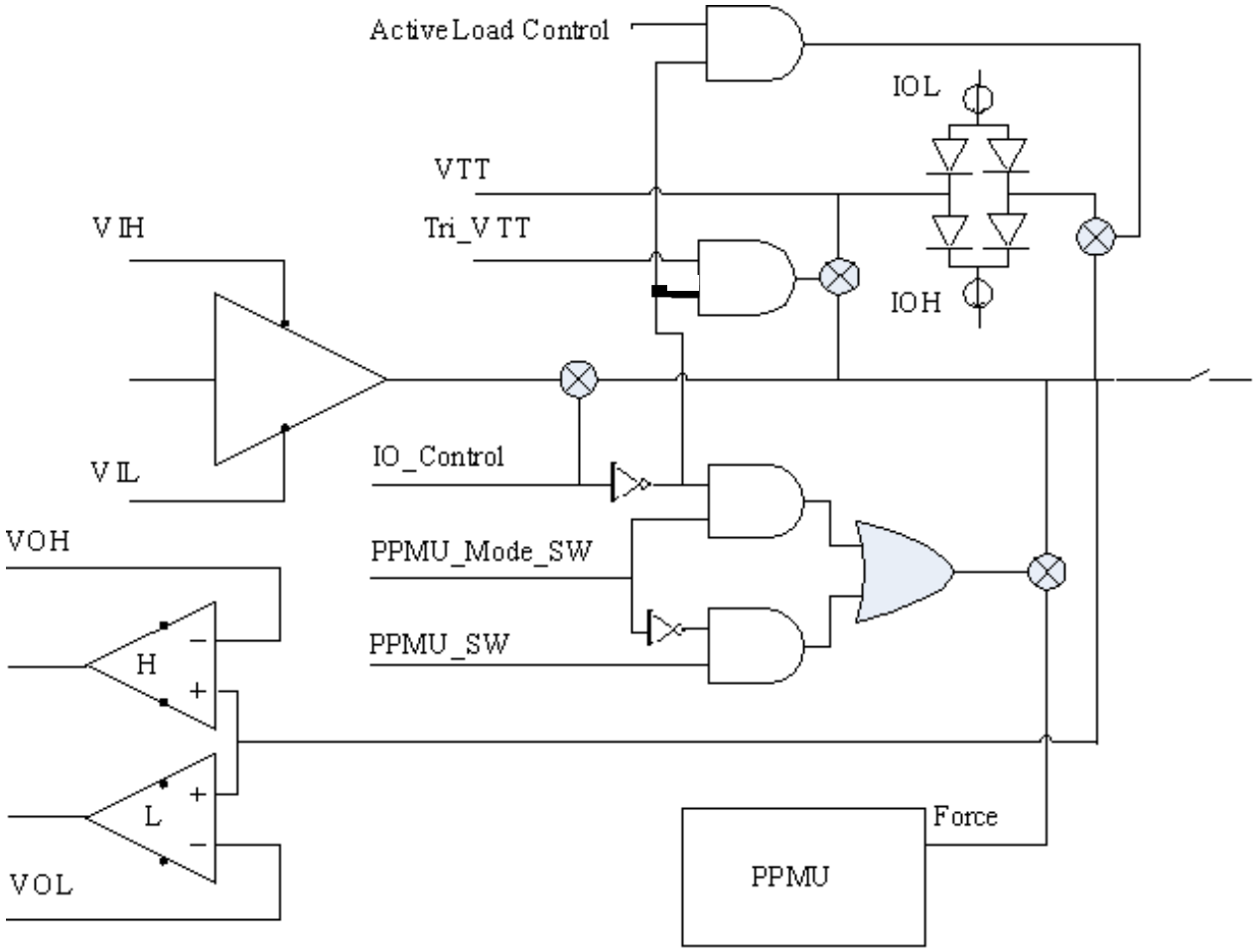
- Pattern generator
  Vector depth: 128/256M
  Log memory depth: 128/256M

- PPMU
  Mode: FV, FI,MV,MI
  Voltage range: -2V~+12V
  Current range: 2uA~32mA

PATTERN GEN BLOCK DIAGRAM



Pin Electronic - Block Diagram



Pin Electronic - Reference Programming

- LEVEL_SET(Level_Set_Name);
  Pin_Name = VIH(Vih_Value), VIL(Vil_Value),VOH(Voh_Value),VOL(Vil_Value);
  :
  :
  END_LS;

- HS_SET_REF_LEVEL(Level_Set_Name);

- ACTIVE_LOAD(Pin_Name, IOH, IOL, Switch);
- SET_VTT(PIN_NAME, Voltage);

Pin Electronic - Timing Programming

- TIMING_SET_GROUP(Timing_Set_Group_Name);
  TS1(Period1) // 10ns~500ns
  {
  Pin_Name{‘Force_Time1’[D/D | D/U | U/D | U/U]; Force_Time2’[D/D | D/U | U/D | U/U]; …… }
  Pin_Name{‘ForceOff_Time' X; ‘Compare_Time1' H/L/Z; ‘Compare_Time2' H/L/Z; …..}
  :
  }

  TS2(Period2) // 10ns~500ns
  {
  :
  }

  :

  TS16(Period16) // 10ns~500ns
  {
  :
  }
  END_TIMING_SET_GROUP;

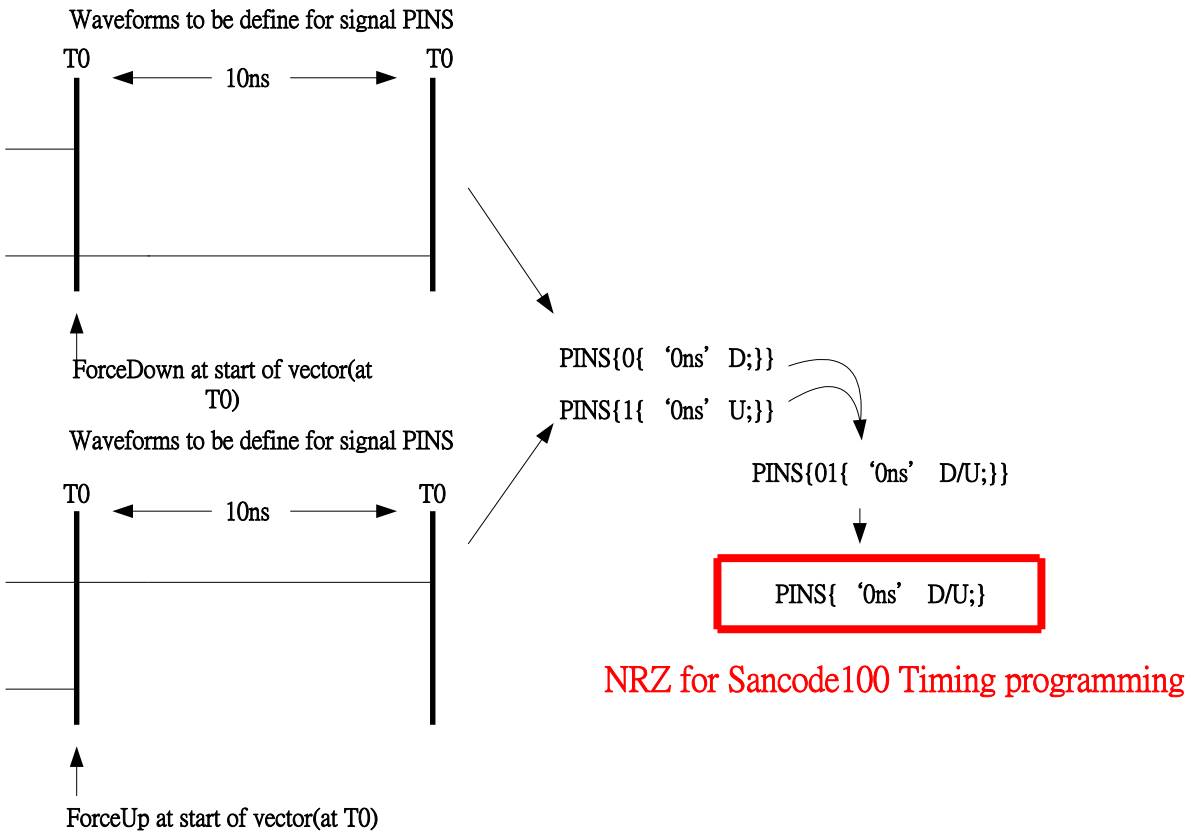
- HS_SELECT_TIMING_SET_GROUP(Timing_Set_Group_Name);

Waveform Programming Description

- There are three components to construct a drive waveform for Sancode system -TIMING_SET, LEVEL_SET & Pattern.
- Waveform format is implied in TIMING_SET.
- Four implied drive & IO waveform format：NRZ、RZ、RTO & SBC.
- Only one implied waveform format in one period, Sancode can repeat the selected waveform format in the same cycle.
- Minimum pulse width 4ns, i.e. maximum clock rate 125MHz.
- Timing skew per pin include drive, IO & strobe.
- Timing resolution 2ns. Skew resolution 20ps.
- Multi-window strobe.

Waveform Programming Description (cont.)

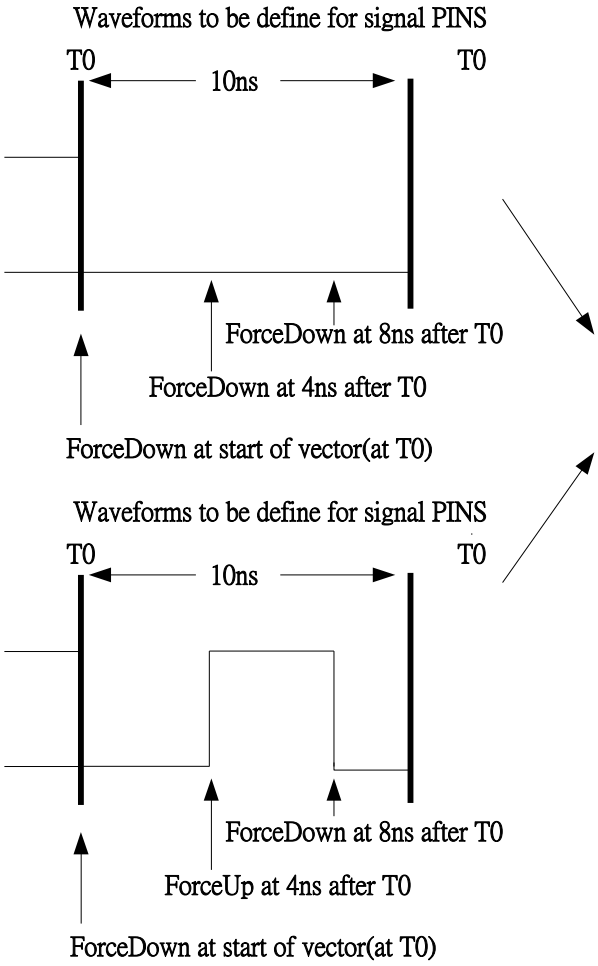
NRZ



NRZ for Sancode100 Timing programming

Waveform Programming Description (cont.)

RZ



PINS{0{ '0ns' D;' 4ns' D;' 8ns' D}}

PINS{1{ '0ns' D;' 4ns' U;' 8ns' D}}

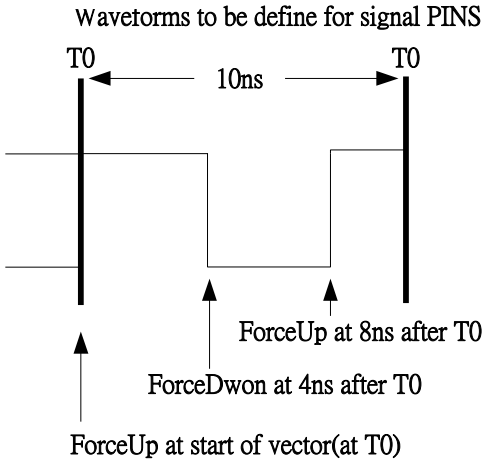
PINS{01{ '0ns' D/D;' 4ns' D/U;' 8ns' D/D}}

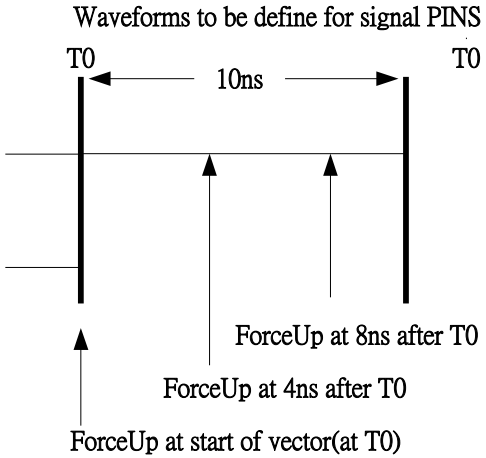
PINS{ '0ns' D/D;' 4ns' D/U;' 8ns' D/D}

RZ for Sancode100 Timing programming

Waveform Programming Description (cont.)

RTO





PINS{0{ ‘0ns’ U;’ 4ns’ D;’ 8ns’ U}}

PINS{1{ ‘0ns’ U;’ 4ns’ U;’ 8ns’ U}}

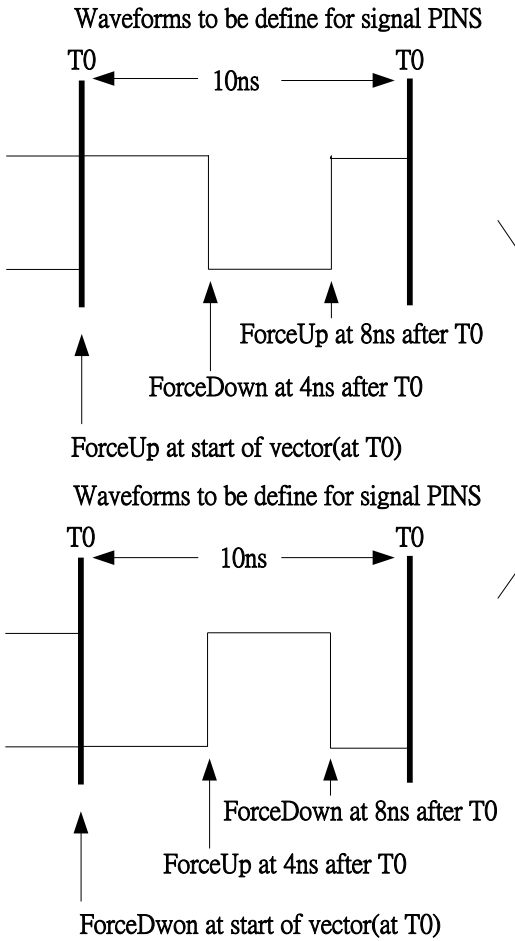
PINS{01{ ‘0ns’ U/U;’ 4ns’ D/U;’ 8ns’ U/U}}

PINS{ ‘0ns’ U/U;’ 4ns’ D/U;’ 8ns’ U/U}

RTO for Sancode100 Timing programming

Waveform Programming Description (cont.)

SBC



PINS{0{ '0ns' U;' 4ns' D;' 8ns' U}}

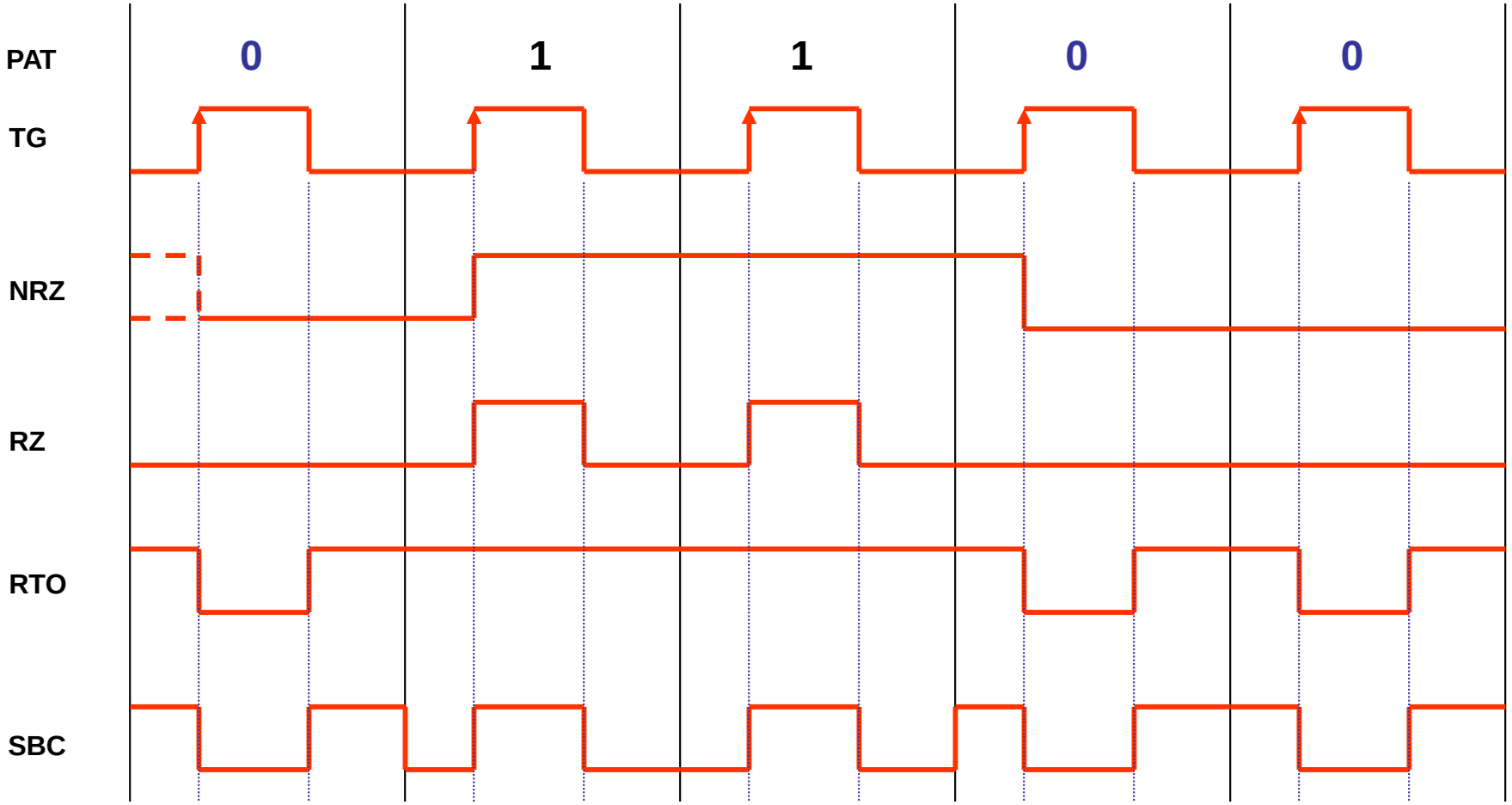
PINS{1{ '0ns' D;' 4ns' U;' 8ns' D}}

PINS{01{ '0ns' U/D;' 4ns' D/U;' 8ns' U/D}}

PINS{ '0ns' U/D;' 4ns' D/U;' 8ns' U/D}

SBC for Sancode100 Timing programming

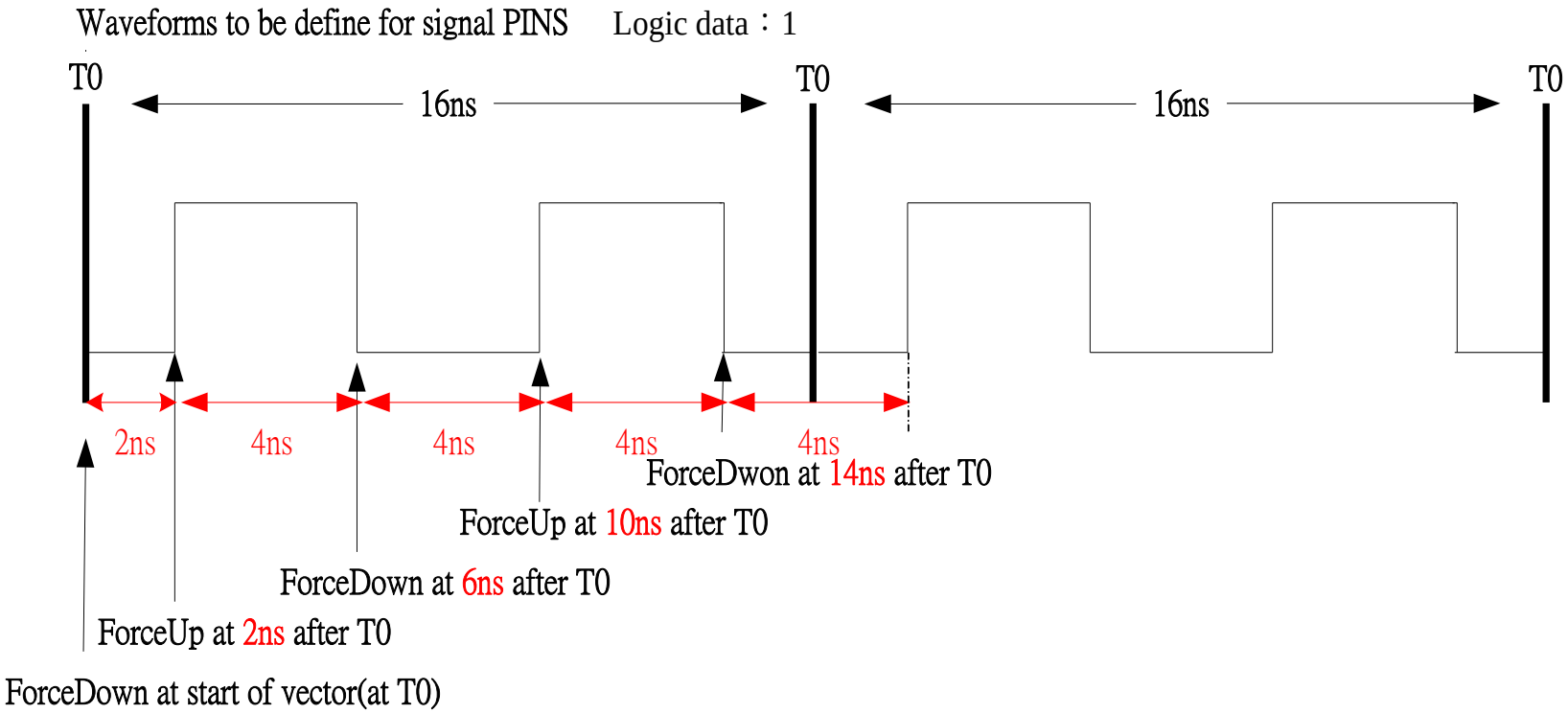
Waveform Programming Description (cont.)



Waveform Programming Description (cont.)

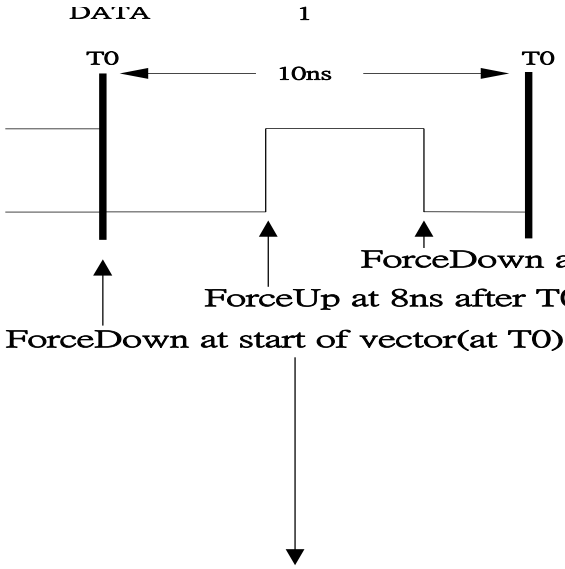
Repeat the selected waveform format in the same cycle

PINS{ ‘0ns’ D/D;’ 2ns’ D/U;’ 6ns’ D/D;’ 10ns’ D/U;’ 14ns’ D/D}



Waveform Programming Description (cont.)

Timing delay

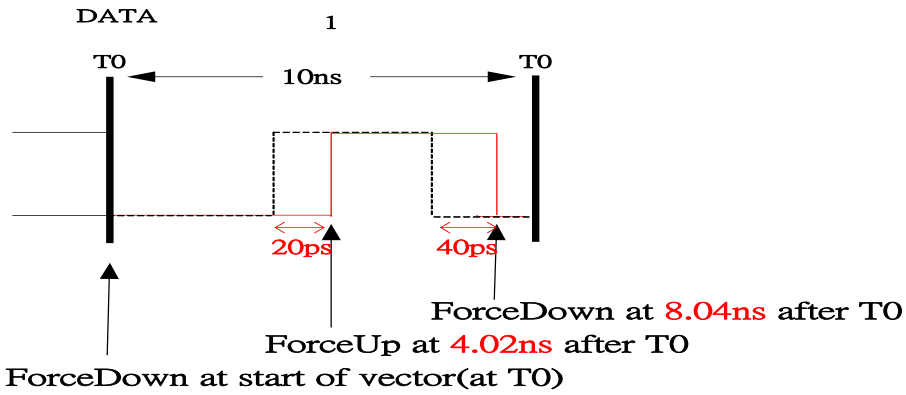


PINS{ ‘0ns’ D/D; ’ 4ns’ D/U; ’ 8ns’ D/D}

Timing delay for Sancode100 timing programming

```
Td
{
    Pin_Group_Name{ ‘Td1’ , ‘Td2’ ;}
}
```

```
Td
{
    PINS{ ‘20ps’ , ‘40ps’ ;}
}
```



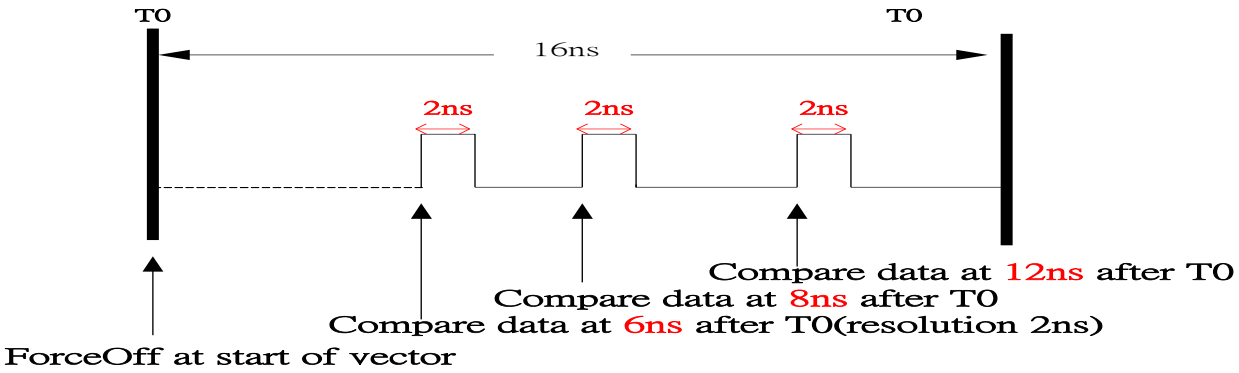
Waveform Programming Description (cont.)

Multi-window Strobe

multi strobe

PINS{ ‘0ns’ X; ’ 6ns’ H/L/Z; ’ 8ns’ H/L/Z;’ 12ns’ H/L/Z}

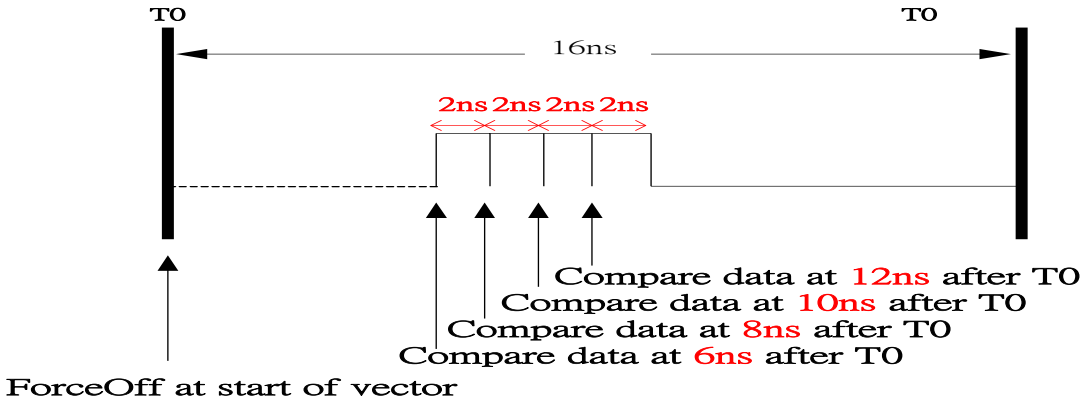
Waveforms to be define for signal PINS



window strobe

PINS{ ‘0ns’ X; ’ 6ns-12ns’ H/L/Z}

Waveforms to be define for signal PINS



PPMU Programming

- **HS_PPMU_FI(Pin_Goupe_Name,Force_Current_Value,I_Range,Clamp_minV,Clamp_maxV, PPMU_Mode_SW,PPMU_SW,Delay_Time);**

- **HS_PPMU_FV(Pin_Goupe_Name,Force_Voltage_Value,I_Range, PPMU_Mode_SW, PPMU_SW,Delay_Time);**

- **HS_PPMU_MI(Pin_Group_Name,Current_Low_Limit,Current_High_Limit, PPMU_Mode_SW,Test_Flag,Delay_Time);**

- **HS_PPMU_MV(Pin_Group_Name,Voltage_Low_Limit,Voltage_High_Limit, PPMU_Mode_SW,Test_Flag,Delay_Time);**

Note: PPMU_Mode_SW options : Normal_Mode or Pattern_Mode.

Test program architecture

- **Two kinds of source files:**

  1: Main test program file.

  2: Vector pattern file.

Test program architecture – Vector pattern file

- **Vector pattern file components:**

1: Pin define:

```
DVC_DEFINE(Device_Name, Package_Type, Pin_Count ,Dut_Count);
    Pin_Name = (Pin_Number, Pin_Type , Tester_Channel);
                    :
 END_DVC_DEFINE;
```

2: Header define:

```
HEADER_DEFINE;
   Header_Name = ( Vector_Bit_1_Pin_Name, Vector_Bit_2_Pin_Name, .....);
                        :
END_HEADER;
```

3: Vector patterns:

```
PATTERN(Pattern_Name, Header_Name);
Label: *01HLZX01HLZX* TS=TSn;
*01HLZX01HLZX *;
          :
          :
*01HLZX01HLZX * Halt;
END_PATTERN;
```

Test program architecture – Vector pattern file (cont.)

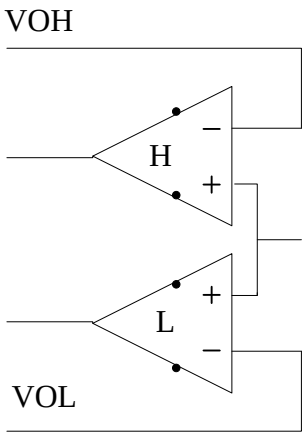
- **Vector pattern symbol:**

Symbol	Note
0	Drive low
1	Drive High
H	Compare High
L	Compare Low
Z	Compare Tri-state
X	No drive & no compare

Test program architecture – Vector pattern file (cont.)

- **Vector pattern log symbol:**

Expect	H			L			Z		
Comparator H state	1	0	0	1	0	0	1	0	0
Comparator L state	1	0	1	1	0	1	1	0	1
Test_Flag_H	0	1	1	1	0	0	1	0	0
Test_Flag_L	0	1	0	1	0	1	0	1	0
Test result	P	F	F	F	P	F	F	F	P
Log Symbol	H	\	+	.	L	-	A	V	Z



Test program architecture – Main test program file

- **Main test program file components:**

1: Pass bin define:

```
PASS_BIN(Pass_Bin_No1, Pass_Bin_No2, .....);
```

2: Pin group define:

```
PIN_GROUP;
   Pin_Group_Name=(Pin1_Name, Pin2_Name, .....);
                :
END_PIN_GROUP;
```

3: Timing set group define:

```
TIMING_SET_GROUP(Timing_Set_Group_Name);;
   TSn(Period)  // 10ns~500ns
   {
          :
   }
        :
        :
END_TIMING_SET_GROUP;
```

Test program architecture – Main test program file (cont.)

4: Reference level define:

```
LEVEL_SET( Level_Set_Name );
      Pin_Name/Pin_Group_Name  = VIH(Vih_Value),VIL(Vil_Value),
                                 VOH(Voh_Value),VOL(Vil_Value);
            :
            :
 END_LS;
```

5: Test plan define:

```
PLAN(Test_Plan_Name);
    // Testing procedure
END_PLAN;
```

6: Test flow & bin define:

```
TEST_FLOW;
   Test_Plan1 = ( Go_Test_Plan2 |  BIN_No, Go_Test_Plan3 |  BIN_No );
                    :
END_TEST_FLOW;
```

Test program & Pattern Compile

- **Pattern compile:**

  PC pattern1 pattern2 pattern3 …..

- **Test program compile:**

  Pre main_test_program_file

Test program – Open/Short (PPMU)

```
#define OSPINS_idx 26
char *OSPINS_GP[OSPINS_idx] = {"A00","A01","A02","A03","A04","A05","A06","A07","A08","A09","A10","A11",
        "A11","A12","A13","A14","IO0","IO1","IO2","IO3","IO4","IO5","IO6","IO7","OE","CE","WE“};

 HS_DPS_FV(DPS1, 0.000V, V2I6, 200mA, ON, ON,1ms);
 PPMU_SENSE_SW(OSPINS, ON);
 HS_PPMU_FV(ALLPINS , 0.0V, I4, Normal_Mode, ON,1ms);
 for(i=0;i<OSPINS_idx;i++)
 {
       Sys_Range[0] = 1;   // V Range
       Sys_Range[1] = 3;   // I Range
       Sys_Flag[0] = 0;    // Mode
       Sys_Flag[1] = 1;    // SV Switch
       $HS_PPMU_FI(-1, OSPINS_GP[i], -0.100, Sys_Range[1], -2000.000, 0.000, Sys_Flag,1);

       WAIT(0.5ms);
       Sys_Flag[0] = 0;   // Mode flag
       Sys_Flag[1] = 1;   // Test Flag
       $HS_PPMU_MV(-1, OSPINS_GP[i], -2000.000, -200.000, Sys_Flag,1);

       Sys_Range[0] = 1;   // V Range
       Sys_Range[1] = 3;   // I Range
       Sys_Flag[0] = 0;    // Mode
       Sys_Flag[1] = 1;    // SV Switch
        $HS_PPMU_FV(-1, OSPINS_GP[i], -0.0, Sys_Range[1], Sys_Flag,1);
 }
 HS_PPMU_FV(ALLPINS , 0.0V, I4, Normal_Mode, OFF,1ms);
```

Test program – Open/Short (Fast)

```
TIMING_SET_GROUP(TG1);
   TS1(500ns)//上限500ns
   {
        ALLPINS{'0ns' D/U;}
        ALLPINS{'0ns' X;'450ns' H/L/Z;}
   }
END_TIMING_SET_GROUP;

LEVEL_SET( LV_OS );
        ALLPINS = VIH(0.500V), VIL(0.00mV),VOH(-0.2V),VOL(-1.2V);
END_LS;

PLAN(OS_Fast)
  HS_DPS_FV(DPS1, 0.000V, V2I6, 200mA, ON, ON,1ms);
  WAIT(5ms);
  HS_SELECT_TIMING_SET_GROUP(TG1);
  HS_SET_REF_LEVEL(LV1);
  PPMU_SENSE_SW(OSPINS, ON);
  HS_PPMU_FV(OSPINS, -2.00V, I4, Pattern_Mode, ON,1ms);
  HS_CONNECT(OSPINS, DCL);
  WAIT(5ms);
  HS_BURST_PATTERN(func_os);
END_PLAN;
```

```
PATTERN(func_os, ALL_PINS);
*OOOOOOOOOOOOOOOOOOOOOOOOOOO* TS=TS1;
*XOOOOOOOOOOOOOOOOOOOOOOOOOO* rpt 20;
*ZOOOOOOOOOOOOOOOOOOOOOOOOOO*;
*OXOOOOOOOOOOOOOOOOOOOOOOOOO* rpt 20;
*OZOOOOOOOOOOOOOOOOOOOOOOOOO*;
*OOXOOOOOOOOOOOOOOOOOOOOOOOO* rpt 20;
*OOZOOOOOOOOOOOOOOOOOOOOOOOO*;
*OOOXOOOOOOOOOOOOOOOOOOOOOOO* rpt 20;
*OOOZOOOOOOOOOOOOOOOOOOOOOOO*;
*OOOOXOOOOOOOOOOOOOOOOOOOOOO* rpt 20;
*OOOOZOOOOOOOOOOOOOOOOOOOOOO*;
      :
      :
*OOOOOOOOOOOOOOOOOOOOOOOXOO* rpt 20;
*OOOOOOOOOOOOOOOOOOOOOOOZOO*;
*OOOOOOOOOOOOOOOOOOOOOOOXO* rpt 20;
*OOOOOOOOOOOOOOOOOOOOOOOZO*;
*OOOOOOOOOOOOOOOOOOOOOOOOX* rpt 20;
*OOOOOOOOOOOOOOOOOOOOOOOOZ*;
*XXXXXXXXXXXXXXXXXXXXXXXXX* Halt;
END_PATTERN;
```

Test program – Leakage

```
PLAN(Leakage)
  HS_DPS_FV(DPS1, 3.300V, V2I6, 200mA, ON, ON,1ms);
  WAIT(5ms);
  HS_SELECT_TIMING_SET_GROUP(TG1);
  HS_SET_REF_LEVEL(LV_CMOS);
  HS_BURST_PATTERN(leakage);

  PPMU_SENSE_SW(OSPINS, ON);
  HS_PPMU_FV(LEAK_PINS, 0.0V, I2, Normal_Mode, ON,1ms);
 for(i=0;i<LEAKPINS_idx;i++)
 {
        Sys_Range[0] = 1;   // V Range
        Sys_Range[1] = 2;   // I Range
        Sys_Flag[0] = 0;    // Mode
        Sys_Flag[1] = 1;    // SV Switch
         $HS_PPMU_FV(-1, LEAKPINS_GP[i], 3000.0, Sys_Range[1], Sys_Flag,1);

        WAIT(0.5ms);
        Sys_Flag[0] = 0;   // Mode flag
        Sys_Flag[1] = 1;   // Test Flag
        $HS_PPMU_MI(-1, LEAKPINS_GP[i], 0.001, -0.001, Sys_Flag,1);

        Sys_Range[0] = 1;   // V Range
        Sys_Range[1] = 2;   // I Range
        Sys_Flag[0] = 0;    // Mode
        Sys_Flag[1] = 1;    // SV Switch
         $HS_PPMU_FV(-1, LEAKPINS_GP[i], 0.0, Sys_Range[1], Sys_Flag,1);
 }
  HS_PPMU_FV(ALLPINS , 0.0V, I4, Normal_Mode, OFF,1ms);
END_PLAN;
```

Test program – Standby Current

```
PLAN(A30_icc_standby);

  HS_DPS_FV(DPS9, 3.300V, V2I6, 200mA, ON, ON,1ms);

  HS_SELECT_TIMING_SET_GROUP(61LV256_TG1);
  HS_SET_REF_LEVEL(LV_CMOS);

  WAIT(5ms);
  HS_BURST_PATTERN(standby);
  WAIT(1ms);
  HS_DPS_MI(DPS9, 0mA, 30mA, ON,1ms);

END_PLAN;
```

Test program – Function

```
PLAN(A50_check_bd_bar);

  HS_DPS_FV(DPS9, 3.300V, V2I6, 200mA, ON, ON,1ms);

  HS_SELECT_TIMING_SET_GROUP(61LV256_TG1);
  HS_SET_REF_LEVEL(LV_CMOS);
  HS_CONNECT(ALLPINS, DCL);
  PPMU_SENSE_SW(ALLPINS, OFF);
  WAIT(5ms);

  HS_BURST_PATTERN(Write_00FF);
  HS_BURST_PATTERN(Read_00FF);

END_PLAN;
```

Test program – IOH/IOL

```
PLAN(A60_VOH);

  HS_DPS_FV(DPS9, 3.300V, V2I6, 200mA, ON, ON,1ms);

  HS_SELECT_TIMING_SET_GROUP(61LV256_TG1);
  HS_SET_REF_LEVEL(LV_CMOS);
  HS_CONNECT(ALLPINS, DCL);
  PPMU_SENSE_SW(ALLPINS, OFF);
  WAIT(5ms);
//**********sink(VOH/IOH)**********
  HS_BURST_PATTERN(Source_1);
  PMU_SENSE_SW(IO, ON);

  HS_PPMU_FI(IO , -4mA,I7, 2.000V, 3.700V, Normal_Mode, ON,1ms);
  WAIT(1ms);
  HS_PPMU_MV(IO , 2.4V, 3.300V, Normal_Mode, ON,1ms);

  HS_PPMU_FV(ALLPINS , 0.0V, I4, Normal_Mode, OFF,1ms);


END_PLAN;
```

Utility – Logic Analyzer

Utility – Logic Analyzer (cont.)

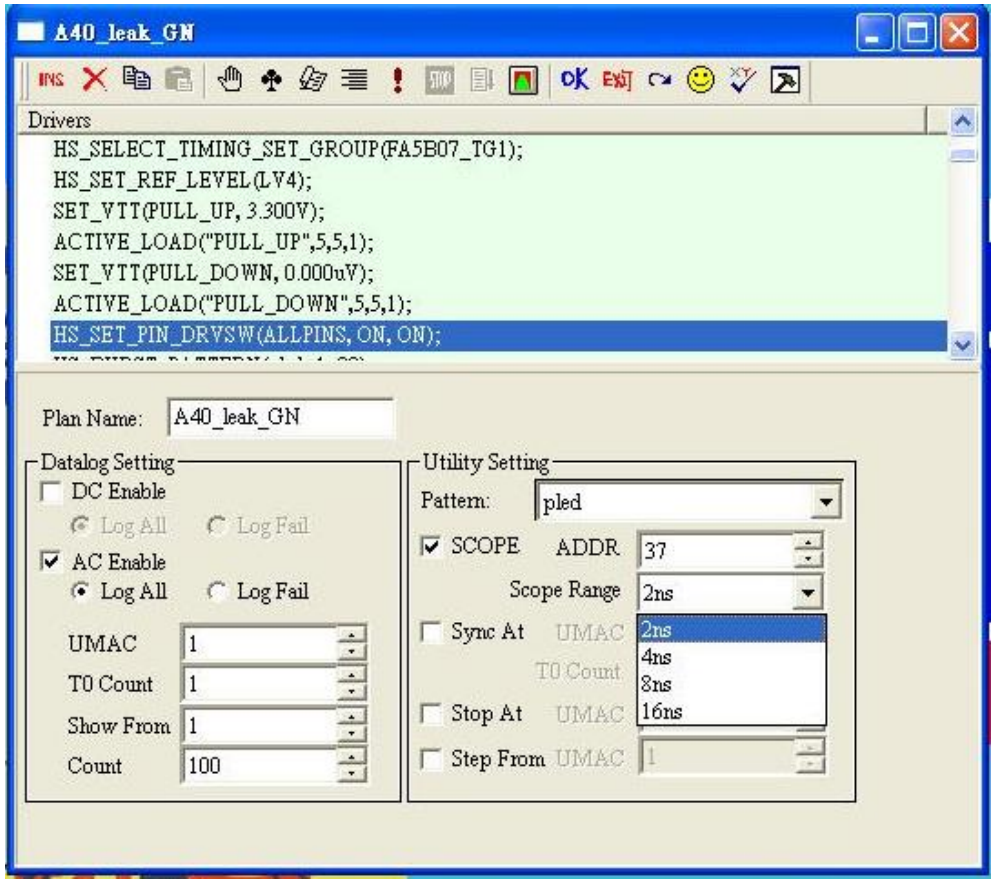
- **Feature**

  1. Real time 2ns strobe resolution.

  2. Per pin capture both for driver and receiver.

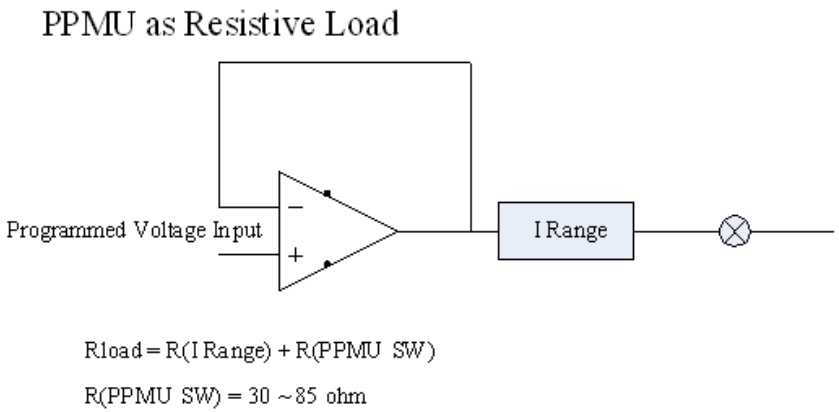
  3. 4K strobe depth.

  4. LA capture, data log & test at the same time.



Application Note

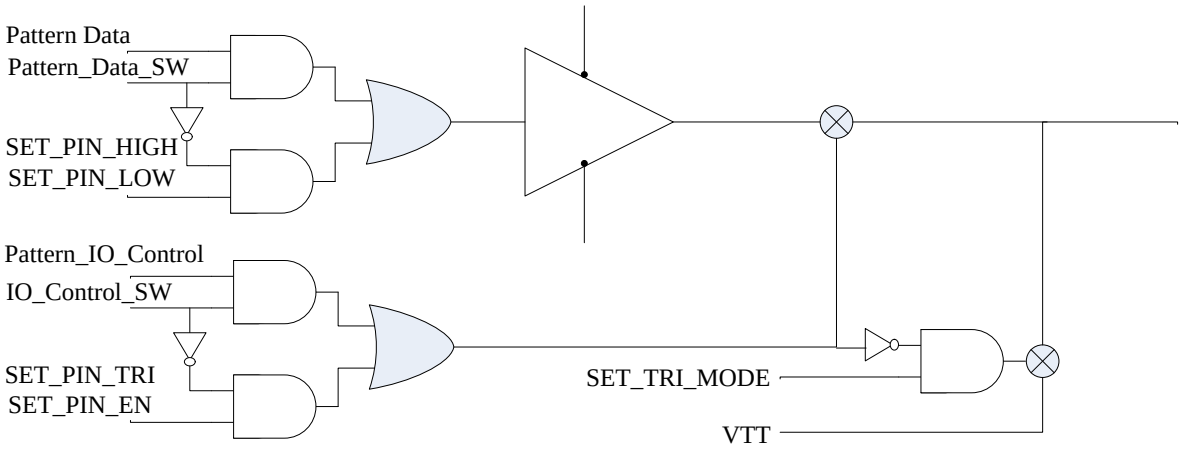
AN 1: PPMU Resistive Load



I Range	Rsense
IR1	500K
IR2	125K
IR3	31.25K
IR4	7.81K
IR5	1.95K
IR6	500
IR7	125
IR8	31.25

- **PPMU_SENSE_SW(PIN, OFF);**
- **HS_PPMU_FV(PIN, Programmed_Voltage, IR_Select, Pattern_Mode, ON);**

AN 2: Pin State Control



- **PIN_INIT(Pin, Pattern_IO_Control, Pattern_Data);**
- **HS_SET_PIN_DRVSW(Pin, Pattern_Data_SW, IO_Control_SW);**
- **HS_SET_PIN_HIGH(Pin);**
  HS_SET_PIN_LOW(Pin);
- **HS_SET_PIN_TRI(Pin);**
  HS_SET_PIN_EN(Pin);
- **HS_SET_TRI_MODE(Pin, Mode); //Mode = SLOW or VTT**