

IEEE P1687 PDL

•Level 0:

- **icomment** <string>
- **ireset**
- **iwrite** <TDR_bit|UCreg|port|pin> <value>
 - lwrite reg|port|pin <name> <value>
 - lwrite [pin] <name> <value> # where name is reg|port if [pin] is omitted
 - lwrite pin::**<name>|<name>** <value> # where pin:: is used only for device-level package pins
- **iread** <TDR_bit|UCreg|port|pin> <value>
- **iapply**
- **irun_loop** tck | sck <int>

- **Maybe not needed for our purposes:**
- **iscope** [pathRelToModule]
- **iscan** <scanPort> <length> -si <siData> -so <soData>

```
"INIT_DATA[1000] (INIT_SETUP)," & -- 1000 bit long INIT_DATA TDR
"INIT_STATUS[500] (INIT_RUN)," & -- 500 bit long INIT_STATUS TDR
```

-- BSDL extensions to describe register with more granularity

attribute REGISTER_NAME of exinit : entity is

```
"IO1[10] (INIT_DATA[110,101]),"&
"IO2[5] (INIT_DATA[210,201]),"&
"status1[1] (INIT_STATUS[100])";
```

-- format register/bus:MNEMONIC (bit pattern, data, data)

attribute WRITE_MNEMONIC of exinit : entity is

```
"Any:IGNORE (XXXXXX)," &
"IO1:off (0000000000)," &
"IO1:2P5 (0000000100)," &
"IO1:PCIE(1100000100)," &
"IO1:3P3 (0000000011),"&
"IO2:off (0000000000)," &
"IO2:2P5 (00100)," &
"IO2:PCIE(1100000100)," &
"IO2:3P3 (0000000011)";
```

attribute READ_MNEMONIC of exinit : entity is

```
"Any:IGNORE (XXXXXXX)," &
"status1:fail(1)," &
"status1:pass(0)";
```

Attribute REGISTER_MNEMONIC

IEEE P1687 examples

```
iComment "SVF like setting of register"
```

```
# clear of init register
```

```
iWrite INIT_REG 0
```

```
# use of user defined sub registers
```

```
iWrite voltage5555 0x555
```

```
# scan the data to the target
```

```
iApply
```

```
iWrite voltage1 0b11011110011
```

```
iApply
```

```
# use of mnemonic rather than number
```

```
iWrite voltage1 PCIE
```

```
iWrite voltage2
```

```
iscan 0x15CA
```

```
iApply
```

```
iRead INIT_STATUS PCIE
```


GUI enables interactive setting of I/O Via hex, binary or MNEMONIC

U1(EXINIT-DW)

PCIE

Pin	Name	To UUT	From UUT
	INSTRUCTION (8)	INIT_SETUP	00
	BYPASS (1)	0	0
	BOUNDARY (18)	01000000000000000000	0XXXXX
	INIT_DATA (1000)	00000000000000000000	000000000000
	INIT_STATUS (500)	00000000000000000000	000000000000
	BCR (2)	0	0
...	Q_0 (8)	00	XX
...	D_I (8)	00	XX
24	OC_NEG_I (1)	1	0
24	Q_C (1)	1	0
1	CLK_I (1)	0	0
	IO1 (10)		000
	IO2 (10)		0000000000
	STATUS1 (1)		PASS

Note Status1 has "Pass" as a return value

IO2 is OFF, preparing to set to PCIe

The screenshot shows a hardware configuration interface with a table of settings and a script editor. The table lists various components and their current states. The 'IO2 (10)' row is highlighted with a black box, and its value is 'OFF'. A mouse cursor is pointing at the 'OFF' value. To the right, a script editor window is open, displaying a sequence of commands: '# assign value', 'iScope U1', 'iComment "set IO2 to PCIe"', 'iWrite IO2 PCIE', and 'iApply'. A blue arrow points to the 'iWrite IO2 PCIE' command.

24	Q_C(1)	1	U		
1	CLK_I(1)	U	U		
	IO1 (10)	P5	000	UX	
	IO2 (10)	OFF	000	UX	
	STATUS1 (1)	U	PASS	PAS	

```
# assign value
iScope U1
iComment "set IO2 to PCIe"
iWrite IO2 PCIE
iApply
```

After iWrite, PCIe is now displayed
iApply will send it to the target

DC_REG_I(1)	1	0	
Q_C(1)	1	0	
1 CLK_I(1)	0	0	
IO1(10)	P5	000	UX
IO2(10)	PCIE	000	UX
STATUS1(1)	0	PASS	PAS


```
# assign value
iScope U1
iComment "set IO2 to PCIe"
iWrite IO2 PCIe
iApply
```


Board or IC level vectors can be recorded

```
ENDDR IDLE;
ENDIR IDLE;
# set IO2 to PCIE
SDR 1000
TDI
(00000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0608000000000000000000000000000000080000000000000000000000000000)
TDO
(000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000)
MASK
(FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF);
```



Same construct but using
Ken Parker's Example

```
attribute REGISTER_NAME of exinit : entity is
    "IO_Bank1[3] (INIT_DATA[16,18]),"&
    "IO_Bank2[3] (INIT_DATA[11,13]),"&
    "IO_Bank3[4] (INIT_DATA[5,8]),"&
    "IO_Bank4[4] (INIT_DATA[0,3]),"&
    "status1[1] (INIT_STATUS[100]);
```

```
-- format register/bus:MNEMONIC (bit pattern, data, data)
```

```
attribute TDI_MNEMONIC of exinit : entity is
    "IO_Bank1:off (000)," &
    "IO_Bank1:3P3V (011)," &
    "IO_Bank2:off (000)," &
    "IO_Bank2:3P3V (011)," &
    "IO_Bank3:off (0000)," &
    "IO_Bank3:Weak_3P3 (1011)," &
    "IO_Bank3:Fast_Slew (1010)," &
    "IO_Bank4:off (0000)," &
    "IO_Bank4:Weak_3P3 (1011)," &
    "IO_Bank4:Fast_Slew (1010)";
```

```
attribute TDO_MNEMONIC of exinit : entity is
    "status1:fail(1)," &
    "status1:pass(0)";
```

U1 (EXINIT-DW)						
WEAK_3P3						
Pin	Name	To UUT	From UUT	Expected	PI	PO
	INSTRUCTION (8)	INIT_RUN	78	81	XXXXXXXXXX	XXXXXXXXXX
	BYPASS (1)	U	0	0	X	X
	BOUNDARY (18)	'UUUUUUUUUUUUUUUU	UXXXX	UXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX
	INIT_DATA (19)	6UUUUU	UXXXX	UXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX
	INIT_STATUS (500)	UUUUUUUUUUUUUUUU	101111000	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX
	BCR (2)	U	U	U	XX	XX
...	Q_0 (8)	UU	XX	XX	XXXXXXXXXX	XXXXXXXXXX
...	D_I (8)	UU	XX	XX	XXXXXXXXXX	XXXXXXXXXX
24	OC_NEG_I (1)	1	U	U	X	X
24	Q_C (1)	1	U	U	X	X
1	CLK_I (1)	U	U	U	X	X
	IO_BANK1 (3)	P3V	U	U	XXX	XXX
	IO_BANK2 (3)	OFF	U	U	XXX	XXX
	IO_BANK3 (4)	OFF	X	X	XXXX	XXXX
	IO_BANK4 (4)	WEAK_3P3	X	X	XXXX	XXXX
	STATUS1 (1)	U	PASS	PASS	X	X

OFF
 WEAK_3P3
 FAST_SLEW

iComment "Setting of IO_Bank 1-4"

```
iWrite IO_Bank1 3P3  
iWrite IO_Bank2 3P3  
iWrite IO_Bank3 Weak_3P3  
iWrite IO_Bank4 Weak_3P3  
iApply
```