

# DSPedia

```
-- instantiate component
component FXPIC1Acc
  generic
    WIDTH : integer := 16;
  port
    in : in std_logic;
    : in std_logic;
    : in std_logic;
    : out unsigned(WIDTH-1 downto 0);
  end port;
end component;

-- local clocks
signal local_clk : std_logic;
signal local_ce : std_logic;

-- local clock enable goes high after first clock enable
signal local_clk_en : std_logic;

-- signals required by the resampled element (integrator/accumulator)
signal resamplerInput : unsigned(WIDTH-1 downto 0);
signal samplerOutput : unsigned(WIDTH-1 downto 0);
signal counter : integer range 0 to RELATIVE_FACTOR-1;
```

## VHDL Reference

```
-- arrays of signals for the resampled element
type RES_T is array (0 to RELATIVE_FACTOR-1) of unsigned(WIDTH-1 downto 0);
signal resamplerInput : RES_T;
signal samplerOutput : RES_T;
```

```
begin

-- connect everything up
AccSecInput <= unsigned(Input);
samplerInput <= AccSecOutput;
CombSecInput <= samplerOutput;

Output <= std_logic_vector(CombSecOutput);
RDY <= output_CLK;
```

```
-- process to generate the decimated input clock
```

```
process (CLK, RST)
begin
  if RST = '1' then
    output_CLK <= '0';
  elsif CE'event then
    if CLK = '1' and CE = '1' then
      input_CLK <= '1';
    else
      input_CLK <= '0';
    end if;
  end if;
end process;
```

```
process (CLK, RST)
begin
  if RST = '1' then
    counter <= 0;
    local_CE <= '0';
```

**steepest ascent**



**DSP Short Course Notes written and developed by:**

**Bob Stewart  
Daniel García-Alís, Garrey Rice, Iain Stirling,  
Graham Freeland, Jamie Bowman**

---

**E-mail: [info@steepestascent.com](mailto:info@steepestascent.com)  
<http://www.steepestascent.com>**

**Steepest Ascent Ltd.  
Ladywell  
94 Duke Street  
Glasgow  
G4 0UW**

**telephone: 0141 552 8855**

---



# Contents:

## 1. Introduction to VHDL

VHDL Basics I	1-1
The Origins of VHDL	1-2
VHDL Basics II	1-3
Benefits of VHDL	1-4
VHDL - Levels of Abstraction I	1-5
Abstraction and Timing	1-6
VHDL in the System Design flow	1-7
VHDL Design Flow	1-8
VHDL Synthesis	1-9
VHDL Tools	1-10
Modelling Hardware in VHDL	1-11
The Design Entity	1-12
Design Entity Example	1-13
Entity Declaration I	1-14
Entity Declaration II	1-15
Architecture Body	1-16
VHDL Keywords	1-17
Libraries and Packages	1-18
VHDL General Points	1-19
AOI with Internal Signals	1-20
Concurrent Signal Assignments I	1-21
Delays in Signal Assignments	1-22
Concurrent Signal Assignments II	1-23
Summary	1-24

## 2. Hierarchy in VHDL

Hierarchy in VHDL	2-1
2-Way Multiplexer	2-2
MUX2 Architecture - AOI Component	2-3
MUX2 Architecture - Inverters	2-4
MUX2 Architecture - Instantiation	2-5
MUX2 Architecture - Multiple Instantiation	2-6
Port Mapping I - Named Association	2-7
Port Mapping I - Positional Association	2-8
Direct Instantiation	2-9
Configuration Specifications	2-10
Entity Binding	2-11
Port Modes	2-12
Processes	2-13
Processes With Sensitivity Lists	2-14
Processes Without Sensitivity Lists	2-15
Test Benches I	2-16
Test Benches II	2-17
Summary	2-18

## 3. Objects and Data Types

Objects in VHDL	3-1
Constants	3-2
Variables	3-3
Signals	3-4
Signals and Variables	3-5
VHDL Types	3-6
Scalar Types (Numeric)	3-7
Scalar Types (Enumerated)	3-8
Defining Your Own Types and Subtypes	3-9
Synthesis of Ints and Enums	3-10
Array Types	3-11
Declaring an Array Type	3-12
Index Range of an Array	3-13
Array Assignments	3-14
Array Literals	3-15
Array Literals Continued	3-16
Records	3-17
Operators	3-18
Tristate and Resolved Signals	3-19
STD_ULOGIC and STD_LOGIC	3-20
STD_LOGIC_VECTOR	3-21
UNSIGNED and SIGNED Types	3-22
Scalar Attributes	3-23
Array Attributes	3-24
Summary	3-25

## 4. Concurrent and Sequential Statements

VHDL Statements	4-1
Concurrent Statements	4-2
Conditional Signal Assignments	4-3
Selective Signal Assignments	4-4
Generate Statements	4-5
For-Generate Statement	4-6
If-Generate	4-7
Sequential Statements	4-8
Signal and Variable Assignments	4-9
If-then-else statements	4-10
If-then-else Synthesis	4-11
Latch Inference	4-12
Case Statements	4-13
For Loops	4-14
For Loop Synthesis	4-15
While Loops	4-16
Infinite Loop	4-17
Summary	4-18

# Contents:

## 5. Simulation and Synthesis

How a VHDL Simulator Works	5-1
Event Driven Simulation	5-2
Event Processing Example I	5-3
Simulation (Delta) Cycles I	5-4
Simulation (Delta) Cycles II	5-5
Delta Cycles - RS Latch Example	5-6
Delta Cycle Race Conditions	5-7
Delta Cycle Race Conditions Example	5-8
Elaboration	5-9
Process Synthesis	5-10
Synthesisable Processes Styles	5-11
Combinational Logic in a Process	5-12
Synthesisable Process Templates	5-13
Bad process Templates	5-14
Summary	5-15

## 6. Finite State Machines

Finite State Machines	6-1
Finite State Machines in VHDL	6-2
Pedestrian Crossing Example	6-3
Traffic Light Controller	6-4
VHDL for the Traffic Light Controller	6-5
Finite State Machine Implementation	6-6
Summary	6-7

## 7. Subprograms and Packages

Subprograms	7-1
Functions	7-2
Properties of Functions	7-3
Function Example	7-4
Function Call Examples	7-5
Procedures	7-6
Properties of Procedures	7-7
Procedure Example	7-8
Procedure Call Example	7-9
Subprogram Declarations	7-10
Where to Declare a Function	7-11
Packages	7-12
Colour Package Declaration	7-13
Colour Package Body	7-14
Summary	7-15

## 8. Hierarchy II

Introduction	8-1
Generic Parameters - Motivation	8-2
Generic Parameters	8-3
Mapping Generics	8-4
Generic Data Path Widths I	8-5
Generic Data Path Widths II	8-6
Configuration Declarations I	8-7
Configuration Declarations II	8-8
MUX2 Example - Default Binding	8-9
MUX2 Example Configuration Declaration	8-10
Assertions	8-11
Summary	8-12