1. INSTANTIATING LPM in Verilog

To promote LPM usage in Verilog design community, this section describes the syntax for instantiating LPM in Verilog design file.

1.1 MODULE DECLARATION

To instantiate component in Verilog design file, the module has to be declared. The declaration of LPM modules is defined in the LPM library file in Support Files section later in this document. In declaring the LPM module, the module should be "fully described". Which includes all the optional ports, parameters. Once defined, any number of these "fully described modules" can be instantiated within the Verilog design. For example, LPM_MULT is declared as the follow:

```
module lpm_mult ( result, dataa, datab, sum, clock, aclr, clken ) ;
    parameter lpm_type      = "lpm_mult" ;
    parameter lpm_widtha    = 1 ;
    parameter lpm_widthb    = 1 ;
    parameter lpm_widths    = 1 ;
    parameter lpm_widthp    = 1 ;
    parameter lpm_pipeline  = 0 ;
    parameter lpm_representation   = "unsigned" ;
    parameter lpm_hint = "unused" ;

    input  clock ;
    input  aclr ;
    input clken ;
    input  [lpm_widtha-1:0] dataa ;
    input  [lpm_widthb-1:0] datab ;
    input  [lpm_widths-1:0] sum ;
    output [lpm_widthp-1:0] result;

      [module body] . . .
    endmodule;
```

1.2 MODULE INSTANTIATION
1.2.1 Port
To avoid the potential mismatch pin connection, LPM required port connection by name. All used port connections needed to be defined in port connect from the instantiation. And the unused optional ports were left out from the port connection and its' default value from the module declaration will be used. For example, the u1 instance in this example only use aset optional port of LPM_FF so other unused optional ports were left our from the port conncetion statement.

```
LPM_FF u1  (.data(result), .q(last), .clock(clock), .aset(clear));
```
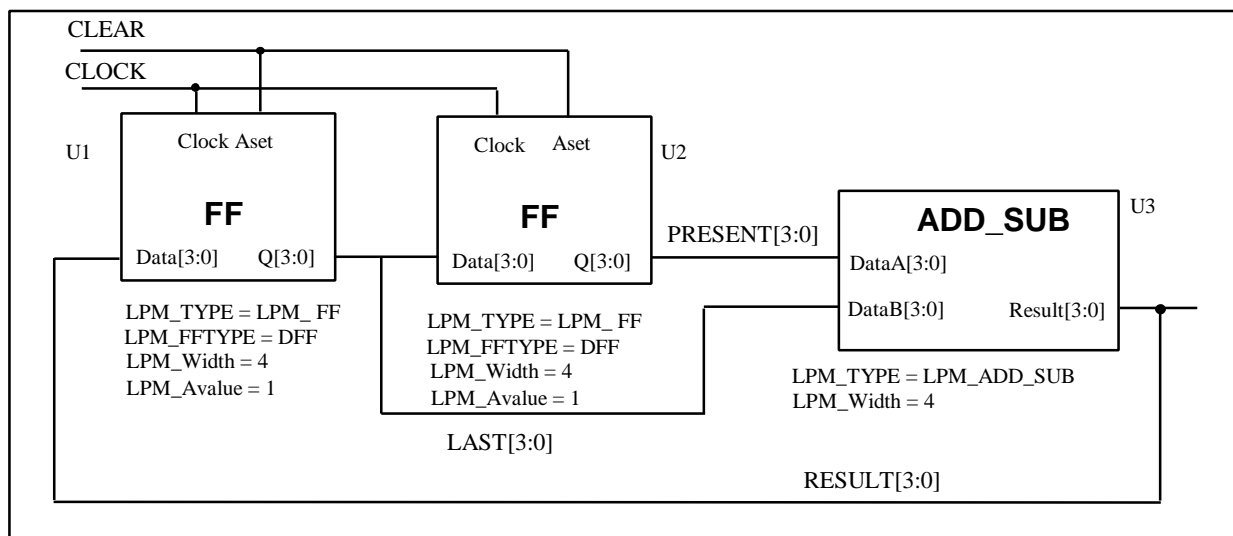
1.2.2 Property

All the used LPM properties values are defined in parameter construct of Verilog. The unused optional properties were left out and its' default value from the module declaration will be used. For example, the u1 instance in this example is a 4-bits DFF with no synchronous set value (ie. LPM_SVALUE) so the LPM_SVALUE is left out from the defparam statements.

```
LPM_FF u1 ...
defparam u1.lpm_width = 4;
defparam u1.lpm_avalue = "0001";
```

1.3 EXAMPLE
1.3.1 Schematic description

The schematic example is shown in the following figure.

1.3.2 Verilog description

```
// Description: LPM instantiation
//
`include "<lpm_module_directory>/lpm_comp.v"

module fibex (clock, clear, result);
input clear;
input clock;
output [3:0] result;
wire [3:0] last;
wire [3:0] present;
```

```
LPM_FF u1 (.data(result), .q(last), .clock(clock), .aset(clear));
    defparam u1.lpm_width = 4;
    defparam u1.lpm_avalue = "0001";

LPM_FF u2 (.data(last), .q(present), .clock(clock), .aset(clear));
    defparam u2.lpm_width = 4;
    defparam u2.lpm_avalue = "0001";

LPM_ADD_SUB u3 (.dataa(present), .datab(last), .result(result));
    defparam u3.lpm_width = 4;

endmodule
```