# RECONFIGURALE HARDWARE DESIGN

www.digitalsystemdesign.in

# Contents

- Field Programmable Gate Array (FPGA) – Reconfigurable hardware for digital design.
- Programmable System-on-Chip (PSoC) - Reconfigurable hardware for mixed signal design.
- Field Programmable Analog Array (FPAA) – Reconfigurable hardware for analog design.

# Exeriments

- FPGA – 6-8 experiments
- PSoC – 2experiments
- FPAA – 2 experiments

# Experiments on FPGA

## *Experiment 1 – Design of Combinatorial circuits.*

1. Design of a 4-bit comparator using 1-bit comparator using structural style.

2. Design of 4-bit array multiplier using structural style. Using 1-bit full adder as the basic block.

## *Home task*

1. 16-bit adder/subtracter
2. 16-bit Mux

# Experiments on FPGA

## *Experiment 2 – Design of Sequential circuits.*

1. Design of a 4-bit loadable counter using structural style.
2. Pipeline implementation of the 4-bit array multiplier.
3. Obtain the design performance parameters for the above multiplier.

## *Home task*

1. 16-bit register.

# Experiments on FPGA

## ***Experiment 3 – Memory Design***

1. Design of a 16X4 Rom using case statement.
2. Design a RAM using behavioral Coding.
3. Introduction to Dual port Memory block design.
4. Introduction the system design using IP cores. Load a memory block with pre-defined data and access the memory block.

# Experiments on FPGA

## *Experiment 4 – State Machines*

1. Basic concepts of state machines using sequence detector. (Moore and Mealy, Overlapping and Non-Overlapping)
2. Serial Adder using FSM.
3. Vending Machine using FSM.

4. Design of UART transmitter and receiver using state machines.

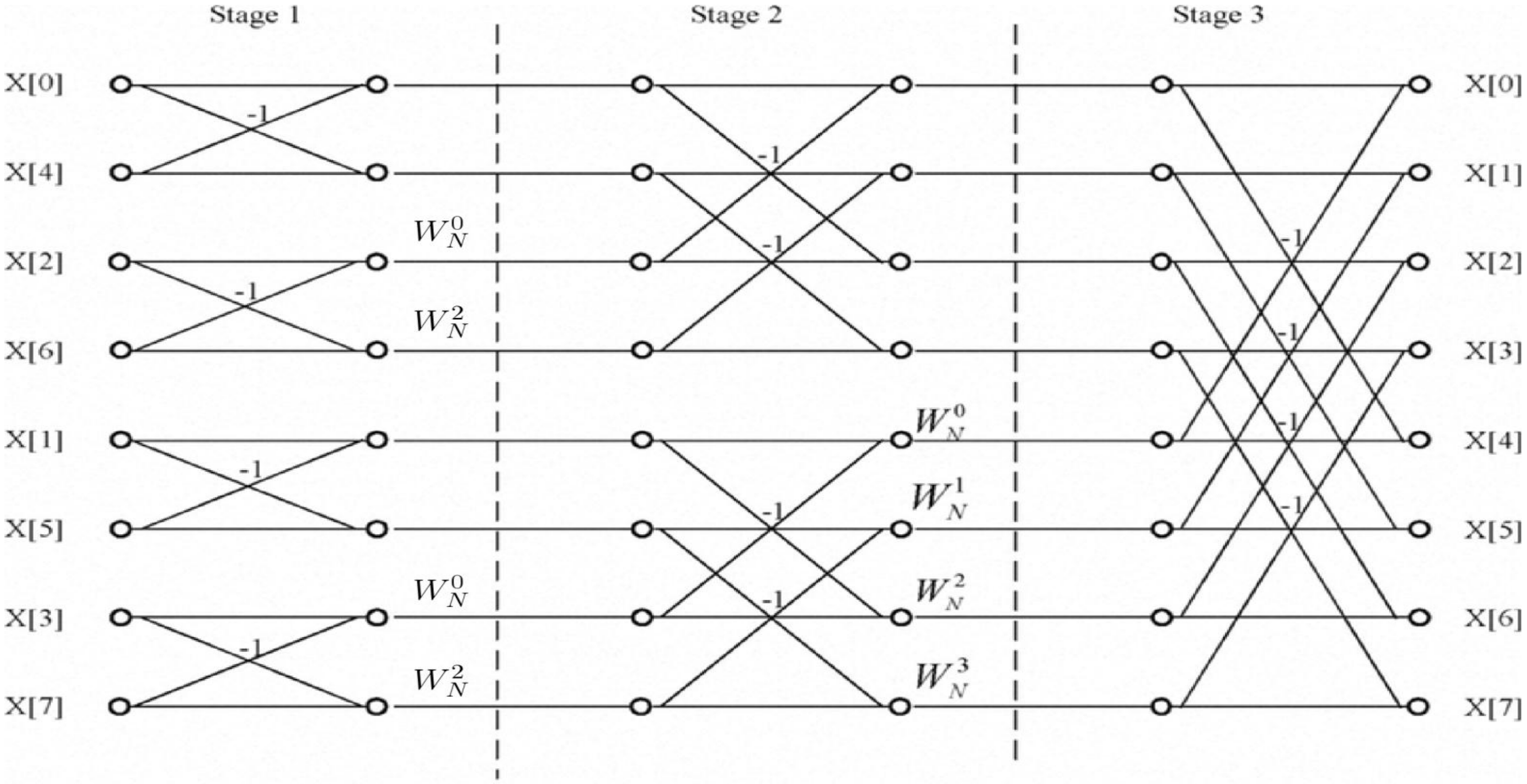
# Experiments on FPGA

Experiment 6 – FPGA implementation of 8-point FFT (Decimation In Time).

1. Design for ASIC - With out any IP cores and DSP blocks
2. FPGA only – With IP cores and advanced DSP blocks

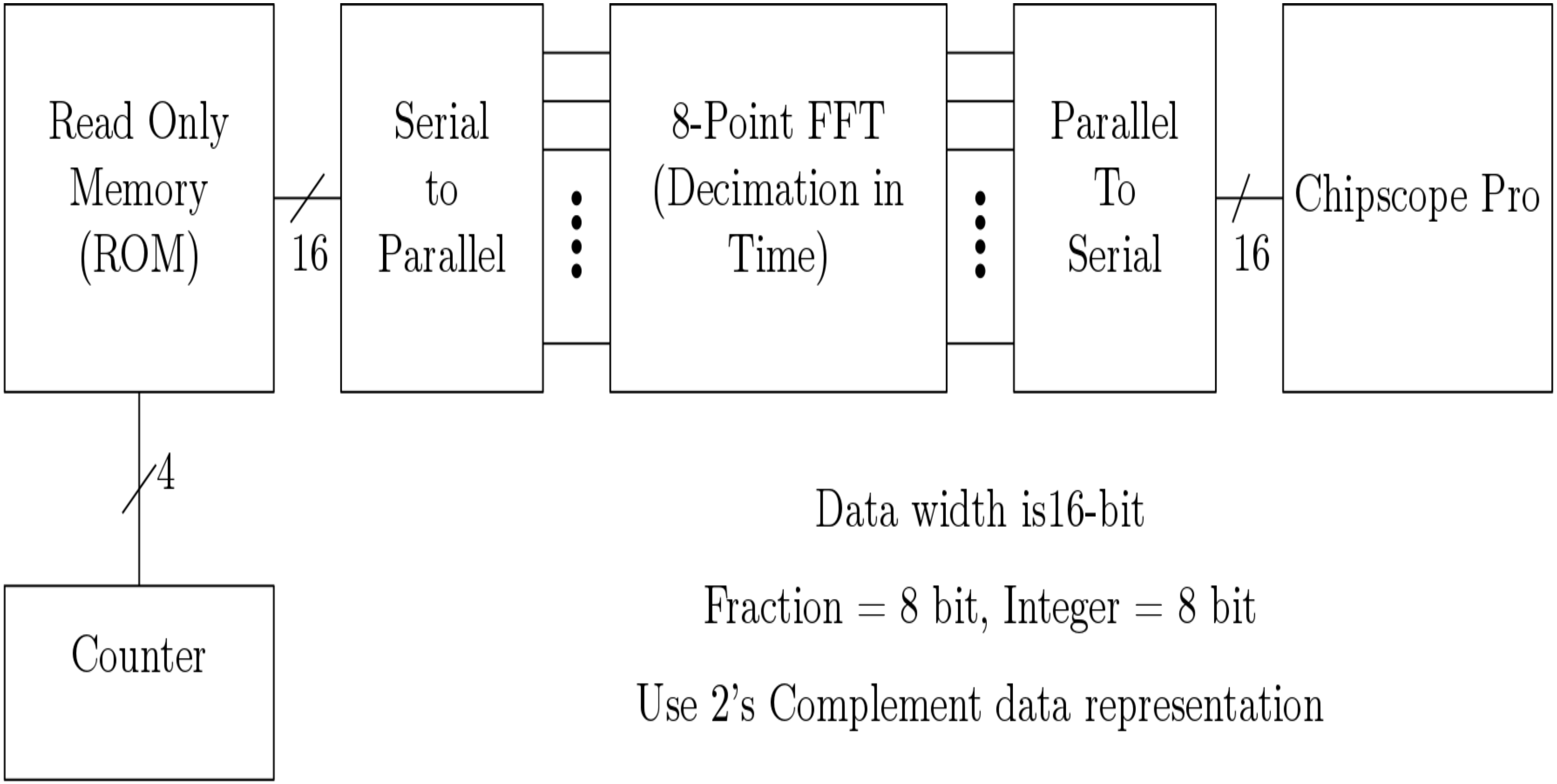
- ➢ Find its design performance (LUT, Register, DSP cores, RAM Blocks, Occupied Slices and Maximum frequency).
- ➢ Analysis of output of the FFT processor using ChipScope pro.
- ➢ Power consumption analysis of the FFT processor using Xpower Analyzer.

# Signal flow for 8-point FFT in DIT



$W_N^0 = 1$ $\quad W_N^1 = \sqrt{2}/2(1-j)$ $\quad W_N^2 = -j$ $\quad W_N^3 = \sqrt{2}/2(-1-j)$

# Experimental Setup for FPGA Implementation of 8-Point FFT

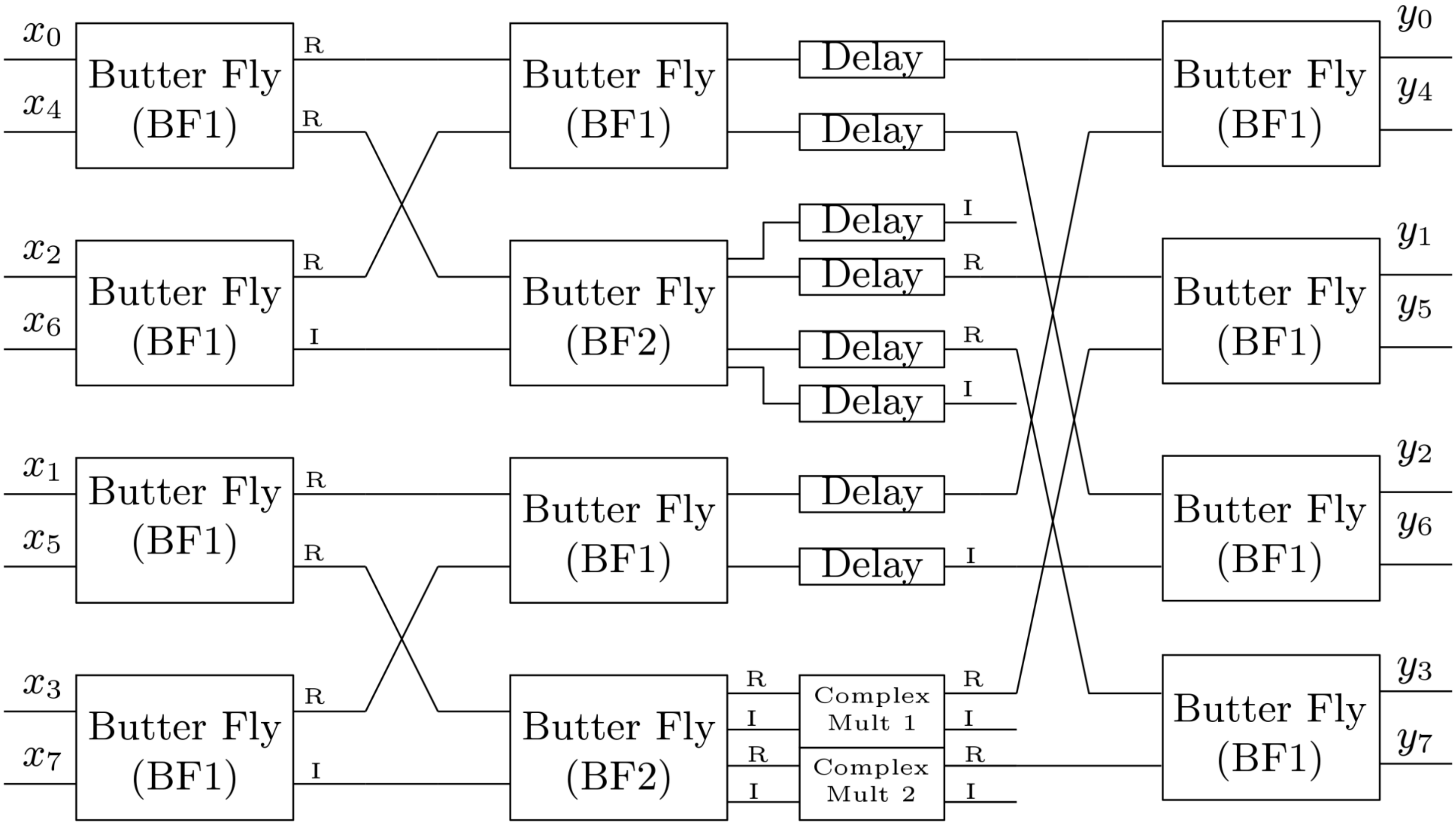


Data width is16-bit

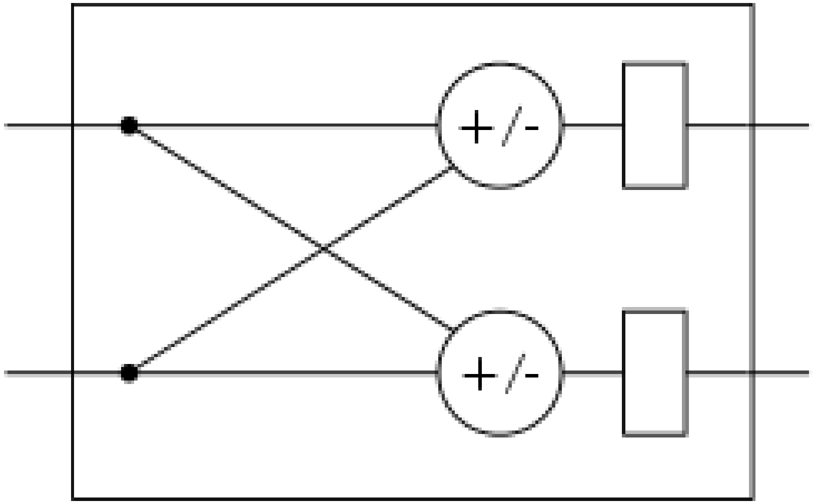
Fraction = 8 bit, Integer = 8 bit

Use 2's Complement data representation

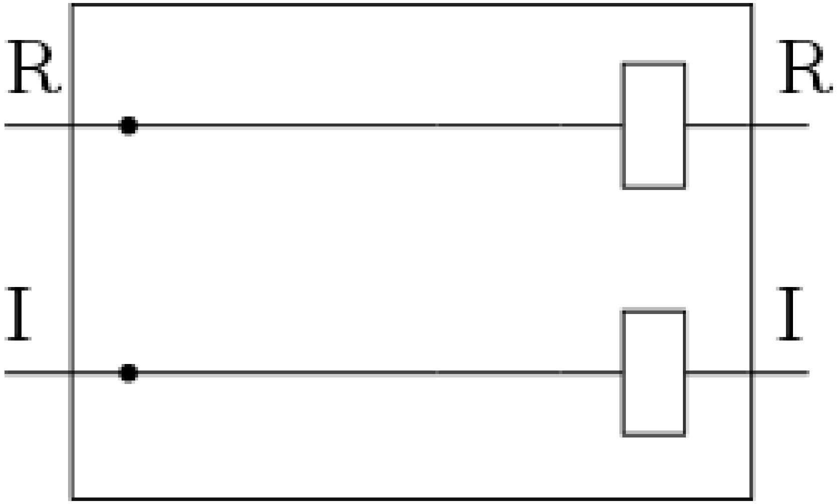
# FFT Architecture



# Basic Butterfly Block

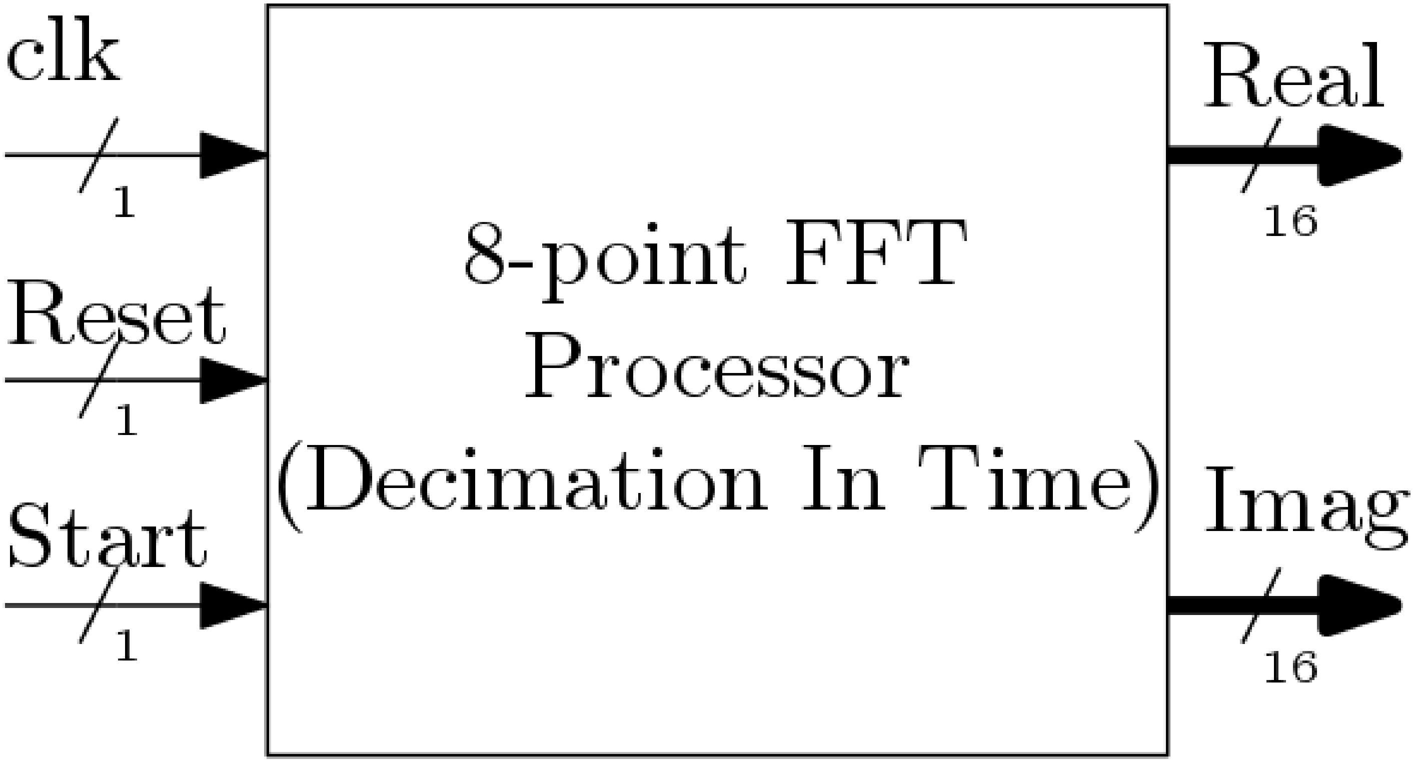


Butter Fly BF1

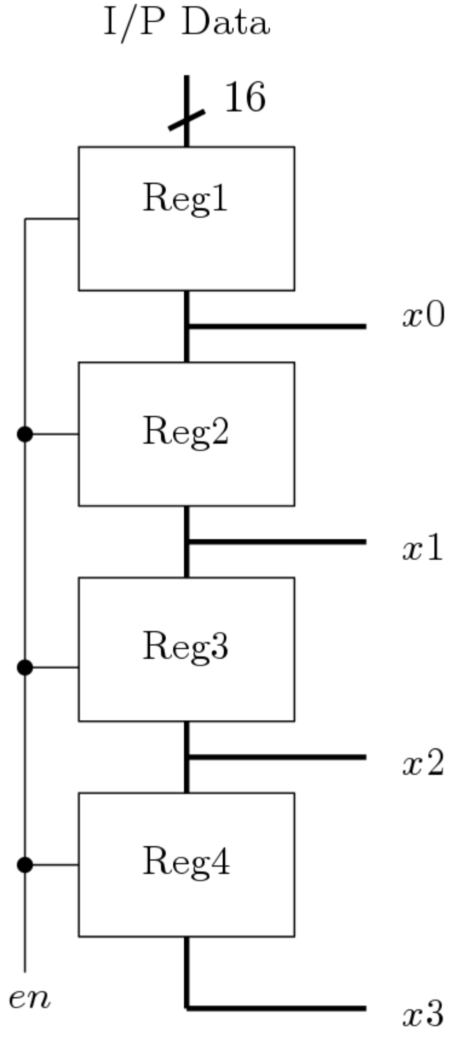


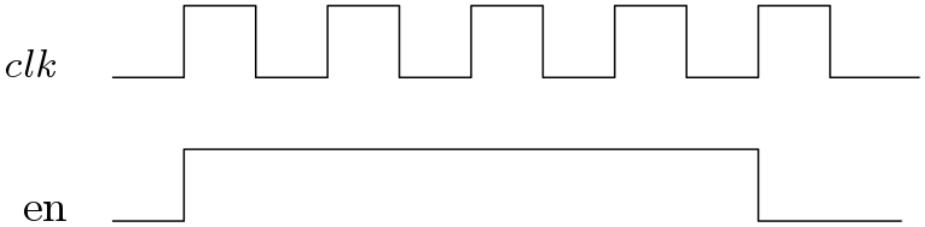
Butter Fly BF 2

# FFT Processor

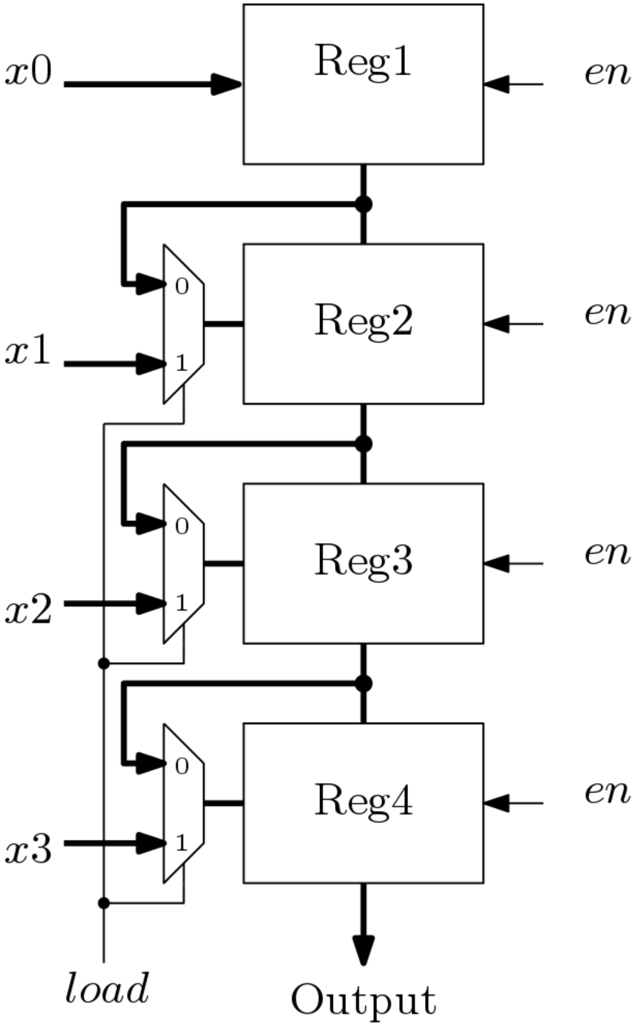


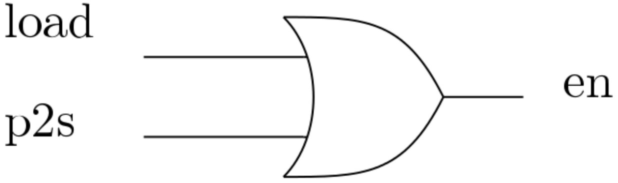
# Serial to Parallel Architecture

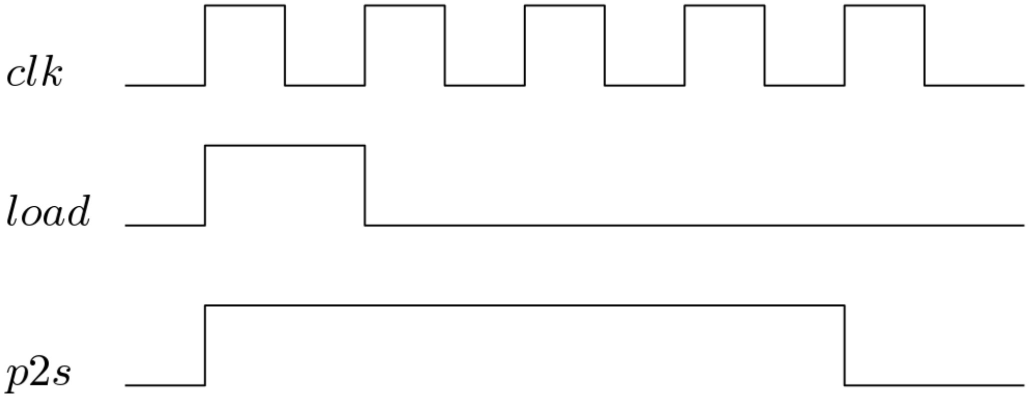




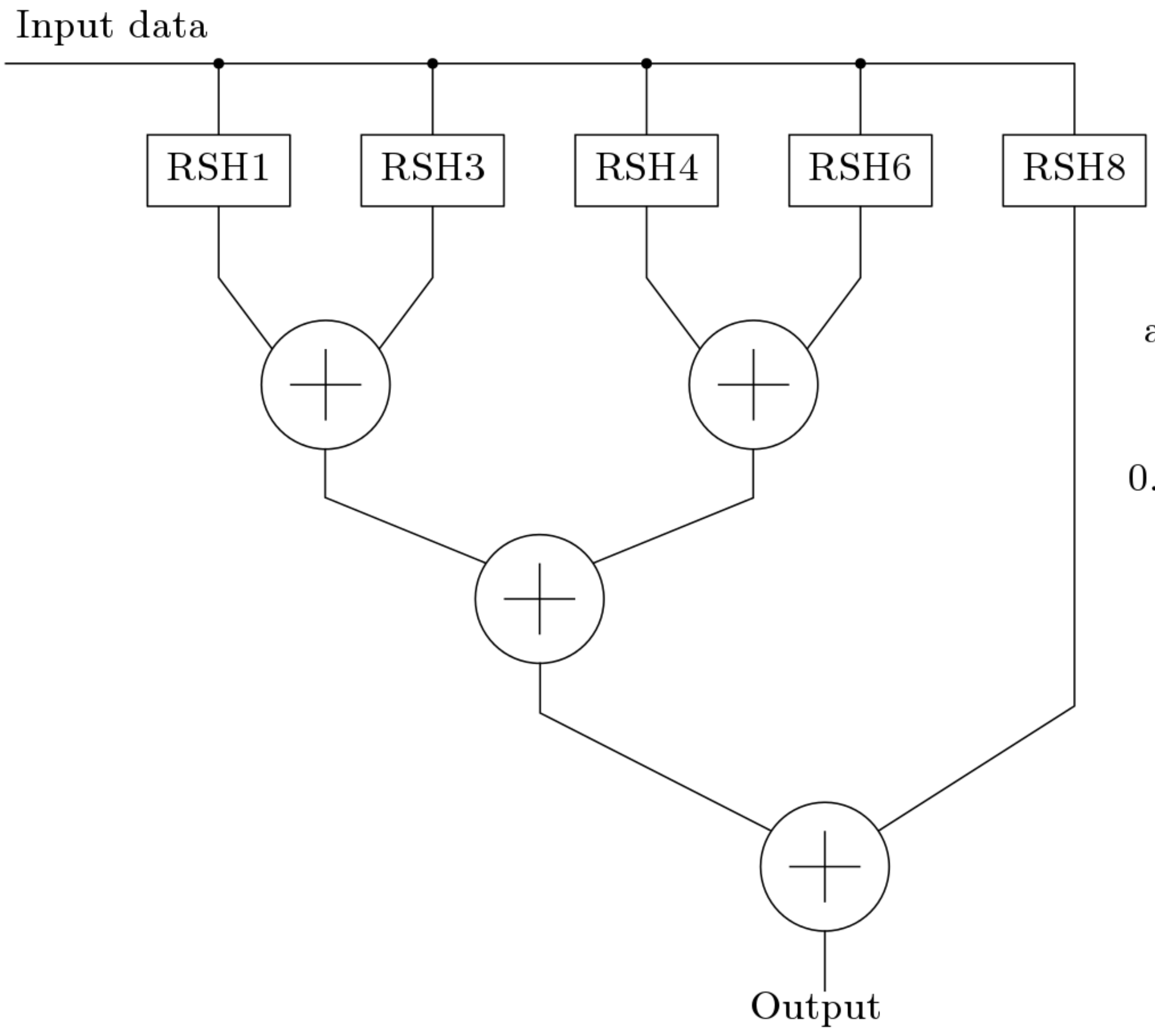
# Parallel to Serial Architecture







# Constant Multiplication Block

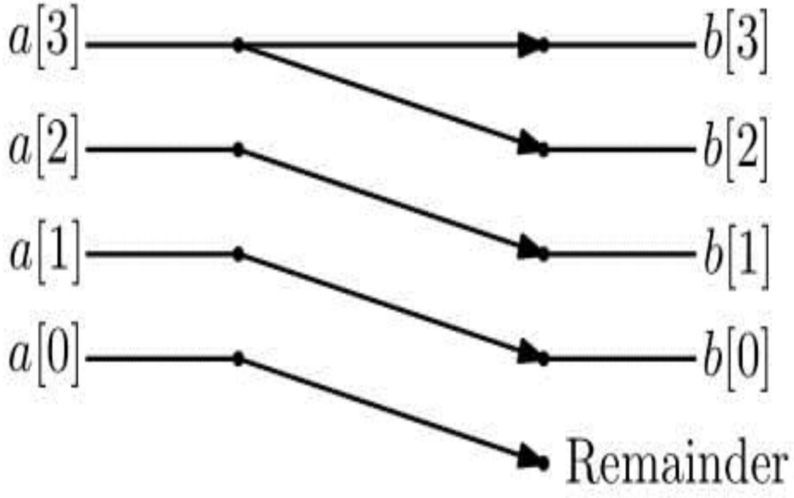


Multiplication by $\frac{1}{\sqrt{2}}$ can be approximated by multplication by 0.7071

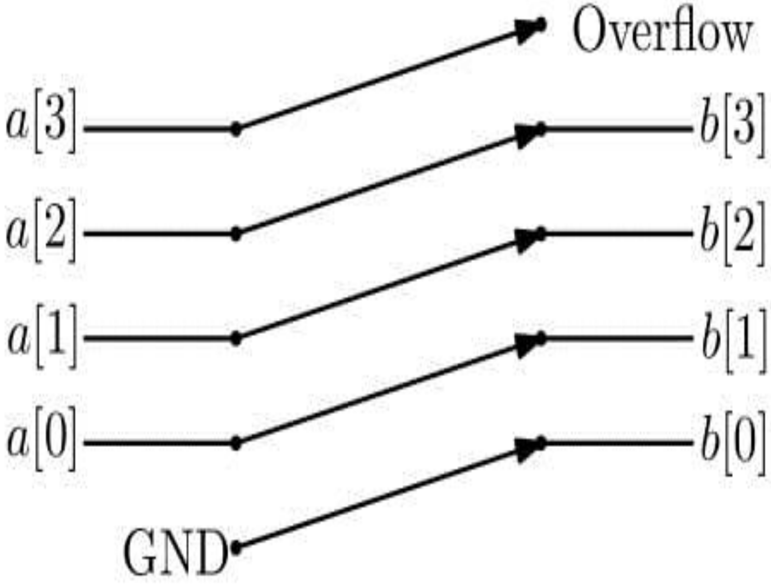
$$0.7071 = 2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-8}$$

The RSH blocks are wired right shift blocks. RSH1 block right shifts the input data by 1 bit.

# Signed Wired Shift Block



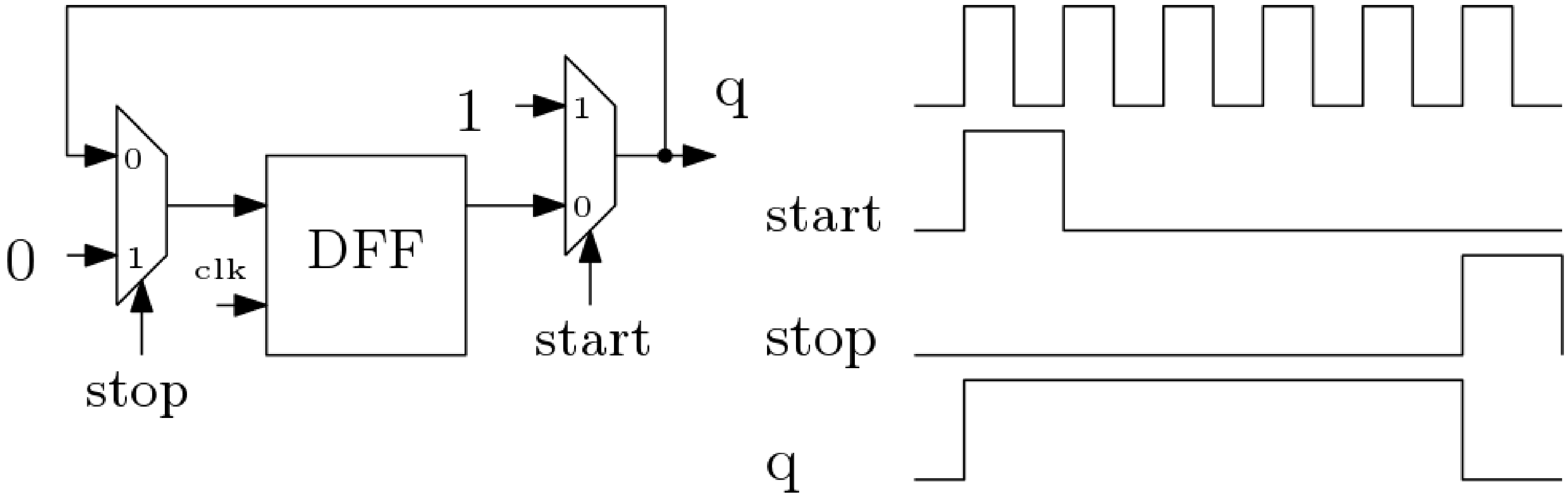
Division by 2 for signed data
(RSH2)



Multiplication by 2 for signed data
(LSH2)

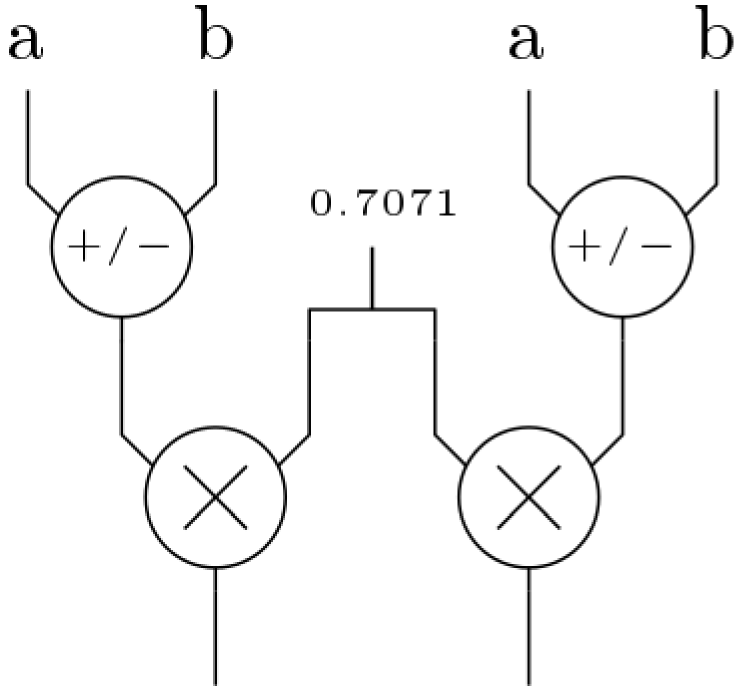
# Control Signal Generation (PG)

- A very simple way to generate control signals is given below.
- Here start and stop pulses are to be controlled by suitable logic and counter blocks.



# Complex multiplier

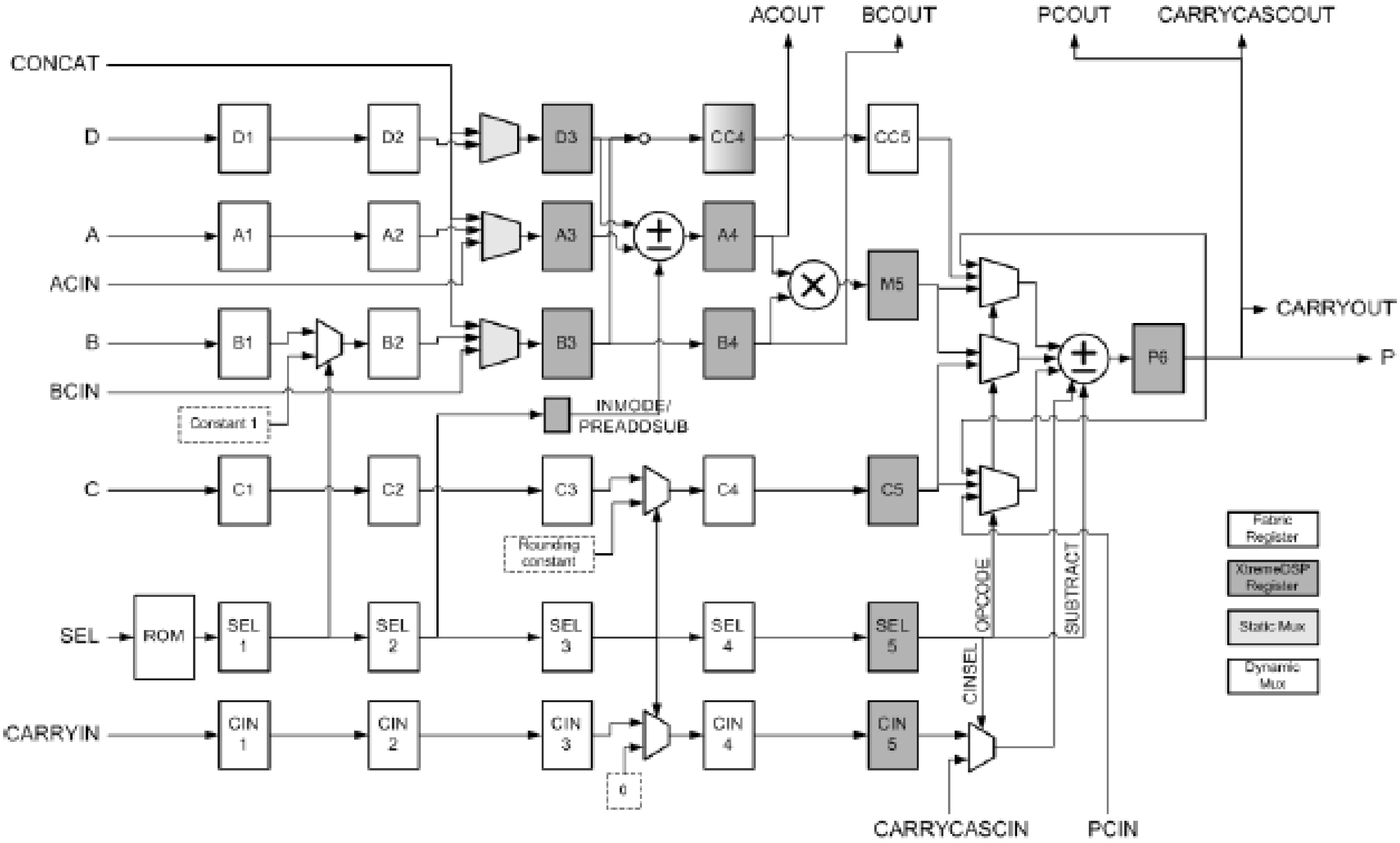
- (a + jb) * (c + jd) = ac + jad + jbc –bd = (ac - bd) + j(ad + bc)
- In our case $w^1$ = 0.7071 – j0.7071, $w^3$ = - 0.7071 – j0.7071
- (a + jb)(0.7071 – j0.7071) = 0.7071(a+b) – j0.7071(a-b)
- (a + jb)(-0.7071 – j0.7071) = -0.7071(a-b) – j0.7071(a+b)



# Required Blocks

1. 16-bit Register
2. 16-bit Adder/Subtractor
3. 16-bit Mux
4. A 8x16 ROM memory element
5. Wired Shift Blocks
6. Constant Multiplier
7. Complex multiplier
8. 4-bit Loadable Up Counter
9. PG blocks

# DSP48 block


ACOUT
BCOUT
PCOUT
CARRYCASCOUT
CONCAT
D
D1
D2
D3
CC4
CC5
A
A1
A2
A3
A4
ACIN
M5
CARRYOUT
B
B1
B2
B3
B4
P6
P
BCIN
Constant 1
INMODE/
PREADDSUB
C
C1
C2
C3
C4
C5
Rounding
constant
Fabric
Register
XtremeDSP
Register
Static Mux
Dynamic
Mux
OPCODE
SUBTRACT
SEL
ROM
SEL
1
SEL
2
SEL
3
SEL
4
SEL
5
CINSEL
CARRYIN
CIN
1
CIN
2
CIN
3
CIN
4
CIN
5
0
CARRYCASCIN
PCIN

# Some Major Warnings and Errors

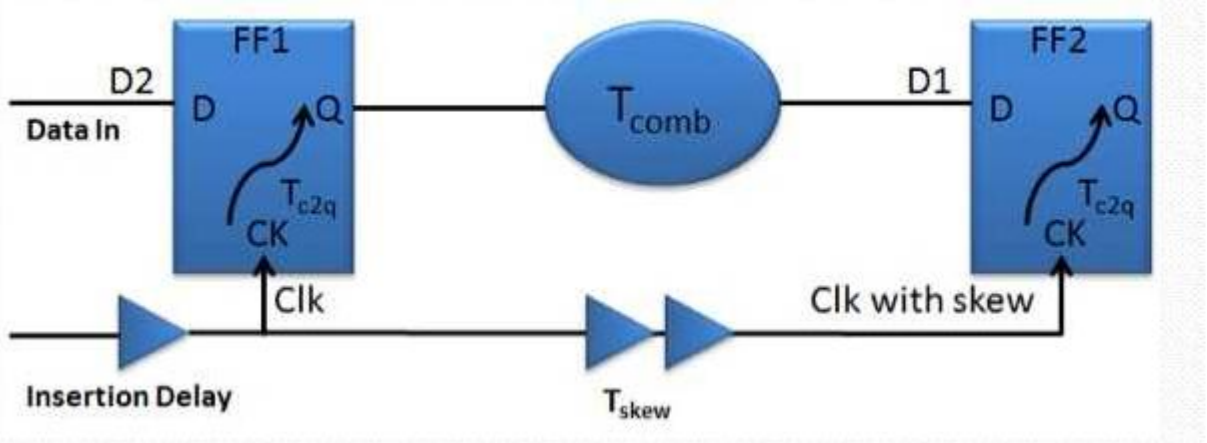
- WARNING:HDLCompiler:1127 - "D:\xilinx_simulation\OMP_ASIC\pg.v" Line 31: Assignment to t2 ignored, since the identifier is never used
- WARNING:Xst:2972 - "D:\xilinx_simulation\OMP_ASIC\pg.v" line 31. All outputs of instance <d2> of block <DFF> are unconnected in block <pg>. Underlying logic will be removed.
- WARNING:Xst:2170 - Unit pg : the following signal(s) form a combinatorial loop: q.
- WARNING:Xst:1290 - Hierarchical block <d2> is unconnected in block <pg>. It will be removed from the design.
- WARNING:Xst:1898 - Due to constant pushing, FF/Latch <d2/q> is unconnected in block <pg>.
- WARNING:Xst:2677 - Node <m28/f1/d1/q> of sequential type is unconnected in block <divider1>.

# Some Major Warnings and Errors

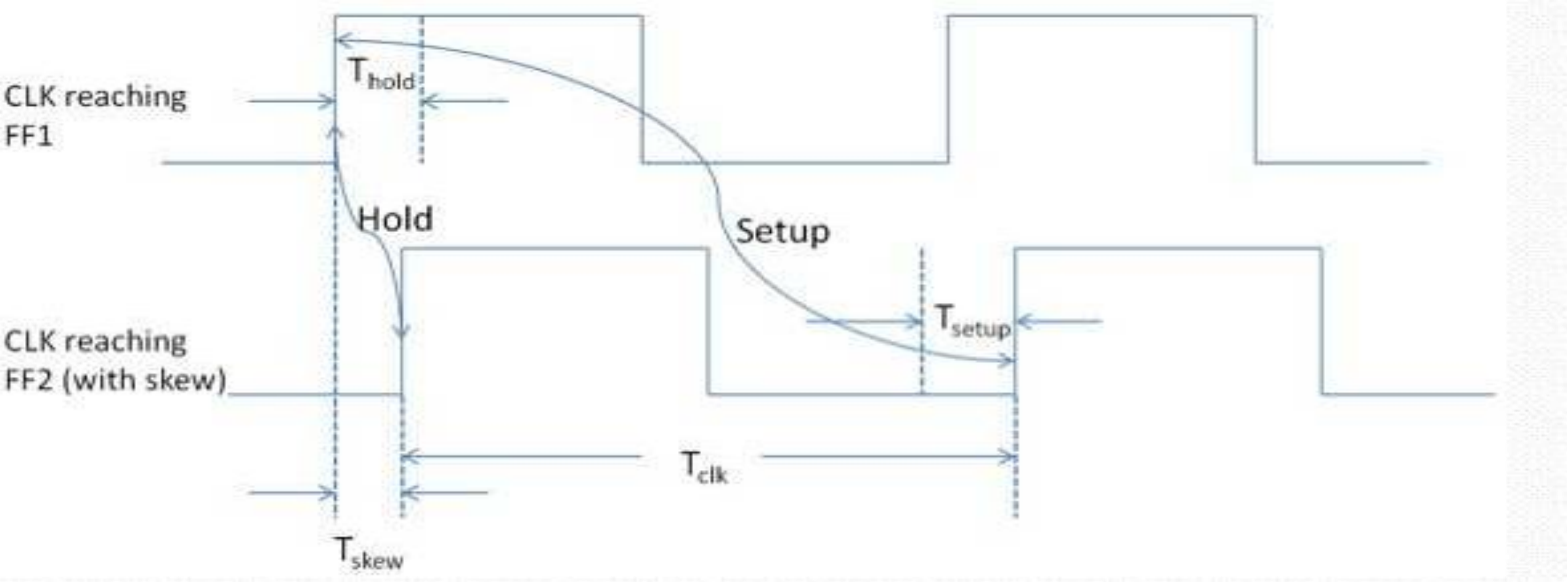
- WARNING:HDLCompiler:189 - "D:\xilinx_simulation\OMP_ASIC\divider1.v" Line 79: Size mismatch in connection of port <c>. Formal port size is 18-bit while actual signal size is 1-bit.
- WARNING:Xst:646 - Signal <b_r> is assigned but never used. This unconnected signal will be trimmed during the optimization process.
- ERROR:Xst:528 - Multi-source in Unit on signal >; this signal is connected to multiple drivers.
- ERROR:Xst:528 - Cannot index into non-array t25
- ERROR:HDLCompiler:329 - "D:\xilinx_simulation\OMP_ASIC\divider1.v" Line 66: Target <t21> of concurrent assignment or output port connection should be a net type.

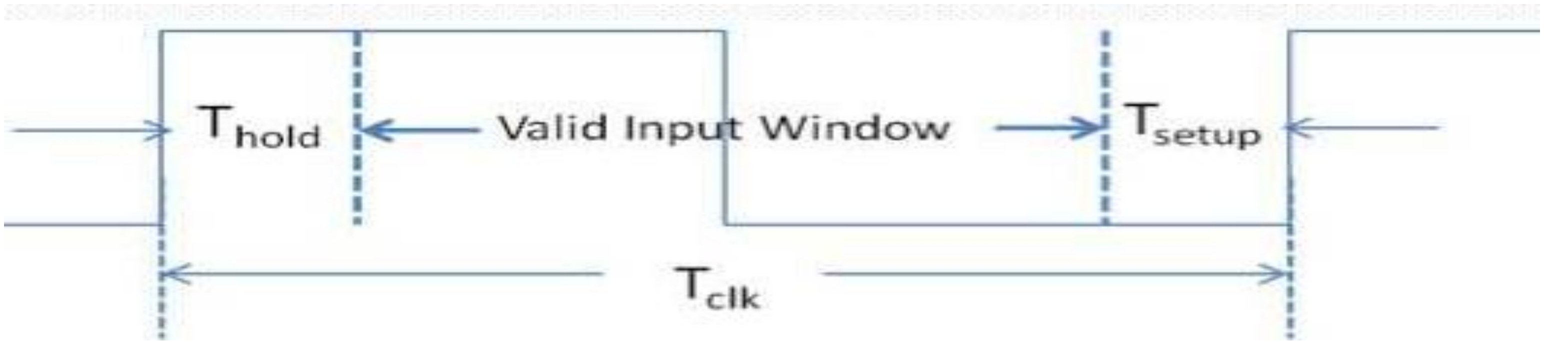
# Setup and Hold Violations Basic

- ❑ In a positive edge triggered flip-flop, input signal is captured on the positive edge of the clock and corresponding output is generated after a small delay called the $T_{c2q}$
- ❑ The clock signal which circulates via clock tree throughout the design has its own variability termed as *skew*



# Setup and Hold Violations Basic





# FFT Example

- Consider the sequence x(n) = {1, 1.5, 2, 2.5,-2.5, -2, -1.5, -1}

**Stage 1**

- {x(0) + x(4), x(0) - x(4), x(2) + x(6), x(2) - x(6),
  x(1) + x(5), x(1) - x(5), x(3) + x(7), x(3) - x(7)}
- {1 – 2.5, 1 + 2.5, 2 – 1.5, 2 + 1.5,
  1.5 – 2, 1.5 + 2, 2.5 – 1, 2.5 + 1}
- x1 = {-1.5, 3.5, 0.5, 3.5, -0.5, 3.5, 1.5, 3.5}
- Involve the constants $w^0$ = 1 and $w^2$ = -j in this stage
- X2 = {-1.5, 3.5, 0.5, 0 - j3.5, -0.5, 3.5, 1.5, 0 - j3.5}
- -(x(3) – x(7)) = x(7) – x(3) similarly for –(x(2) – x(6))
- For real x2(3) and x2(7) is zero.

# FFT Example

**Stage 2**

- X2 = {-1.5, 3.5, 0.5, 0 - j3.5, -0.5, 3.5, 1.5, 0 - j3.5}
- X3 = {x2(0) + x2(2), x2(1) + x2(3), x2(0) - x2(2), x2(1) - x2(3), }
  x2(4) + x2(6), x2(5) + x2(7), x2(4) - x2(6), x2(5) - x2(7),}
- X3 = {-1.5 + 0.5, 3.5 – j3.5, -1.5 – 0.5, 3.5 + j3.5,
  -0.5 + 1.5, 3.5 – j3.5, -0.5 -1.5, 3.5 + j3.5}
- X3 = {-1, 3.5 – j3.5, -2, 3.5 + j3.5, 1, 3.5 – j3.5, -2, 3.5 + j3.5}
- Involve the constants $w^0, w^1, w^2, w^3$ in this step.
- X4(4) = x3(4) * $w^0$ = (1) * (1) = 1
- X4(5) = x3(5) * $w^1$ = (3.5 – j3.5) * (0.7071 – j0.7071) = 0 – j4.949
- X4(6) = x3(6) * $w^2$ = (-2) * (-j) = j2
- X4(7) = x3(7) * $w^3$ = (3.5 + j3.5) * (-0.7071 – j0.7071) = 0 – j4.949

# FFT Example

## Stage 3

- X4 = {-1, 3.5 – j3.5, -2, 3.5 + j3.5, 1, -j4.949, j2, -j4.949}
- y = {x4(0) + x4(4), x4(1) + x4(5), x4(2) + x4(6), x4(3) + x4(7),

  x4(0) – x4(4), x4(1) - x4(5), x4(2) – x4(6), x4(3) – x4(7)}
- y = {-1 + 1, 3.5 – j3.5 - j4.949, -2 + j2, 3.5 + j3.5 - j4.949 ,

  -1 - 1, 3.5 – j3.5 + j4.949, -2 - j2, 3.5 + j3.5 + j4.949 ,
- y = {0, 3.5 – j8.449, -2 + j2 , 3.5 – j1.449,

  -2, 3.5 + j1.449, -2 - j2, 3.5 + j8.449}

# Setup and Hold Violations Basic

- Setup Time: the amount of time the synchronous input (D) must be stable before the active edge of the clock.

$$T_{c2q} + T_{comb} + T_{setup} \leq T_{clk} + T_{skew}$$

- Hold Time: the amount of time the synchronous input (D) must be stable after the active edge of the clock

$$T_{c2q} + T_{comb} \geq T_{hold} + T_{skew}$$

- To avoid the setup violation to achieve a particular frequency insert pipeline registers. This will increase latency.
- Hold violations are corrected by XILINX itself by inserting intermediate buffers.
- For power analysis, all the constraints should be met by fulfilling all setup and hold violations. Design should be completely routed.

# Experiments On FPAA

1. Introduction to the Anadigm FPAA kit. Realize a basic Gain amplifier and simulate using Anadigm Designer 3.
2. Realization of basic rectifier circuits and simulation using Anadigm Designer 3.
3. Design an inverting or non-inverting amplifier and simulate using Anadigm dual apex board. Estimate resource utilization.
4. Realization of the basic filter circuits with or without inbuilt filter FPAA functions. Program the FPAA and observe the output.
5. A sinusoidal input is to be modulated by an arbitrary square wave of higher frequency (Carrier Signal). The arbitrary square wave should be given from an LUT. Output is to be seen by CRO.

# Experiments on PSOC

1. Introduction to PSOC3 development kit and WAP to blink LED with pulse width modulation on PSOC3 kit using CY8C38.
2. Implement Delta Sigma ADC in Differential Mode to display digital value on LCD module of fed analog voltage with PSoC3 kit using CY8C38.
3. Implement Analog Voltage Comparator with PSoC3 kit using CY8C38.
4. Implement 7-bit Down Counter with PSoC3 kit using CY8C38.
5. Implement inverting programmable gain amplifier with PSOC3 using CY8C38.