Programmable Digital Signal Processors (pDSP) and Field Programmable Gate Arrays (FPGA)

pDSP

- Most commonly used DSP solution
- **Basic Characteristics**
  - High Speed Arithmetic
  - Excellent Real Time Signal Handling
  - Multiple Memory Access
- **High Speed Arithmetic**
  - Multiply and Accumulate (MAC) in one cycle
  - Multiple MAC Units

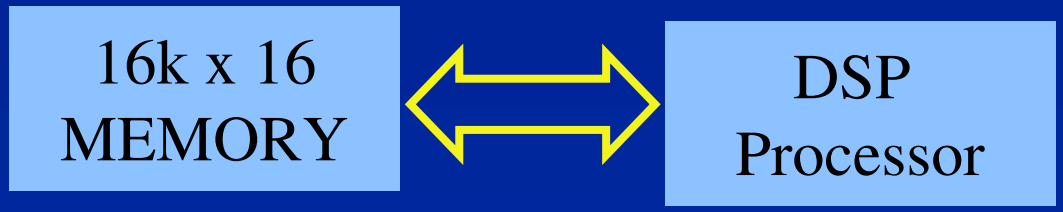
pDSP

Multiple Memory Access

Harvard Architecture

Modified von Neuman Architecture

Memory clock runs faster than Instruction Cycle



pDSP

Excellent at real time signal handling

- Various types of Port Support
  - *Serial Port:* High Speed Synchronous or Asynchronous Serial Port
  - *Host Port:* Communication with Hosts like PC
  - *Link (or Comm) Port:* Interprocessor Communication
- Data can be transferred without stopping the DSP

pDSP

Signal In / Out

System Controller

Other DSP Processors

DMA

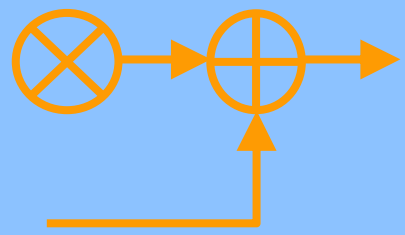
EFFICIENT I/Os

REGISTERS
- Integer
- Float

PORTS
- Serial
- Host
- Link

ADDRESS Generator

PARALLEL MAC

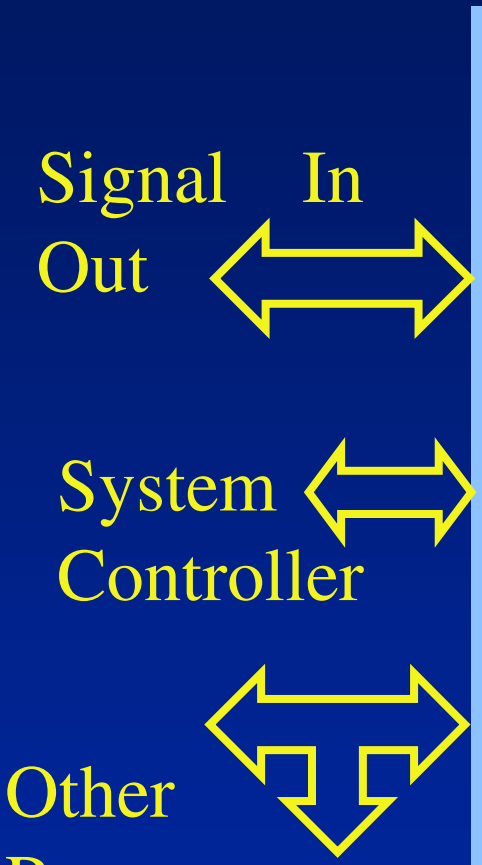


MULTIPLE MEMORIES

16k x 16 PROGRAM

16k x 16 DATA

Virtex FPGA



EFFICIENT I/Os

Register Rich

Distributed MAC & Logic

Drop In Comm. Protocols

Drop In Cores

Constant or Variable Multipliers

Fast External Memory Interface

16 or 32 Deep LUTs and Memories

Fast Carry Logic

MEMORIES

Block Memory

Distributed Memory

Virtex FPGAs

High Speed Arithmetic

- Use Distributed Arithmetic Technique which utilizes the Look Up Tables, Distributed Synchronous RAM Units and Fast Carry Logic to do the multiplies
- Use Multiple Constant or Variable Multipliers
- Decide on the number of parallel units from the required sample rate

Virtex FPGAs

Input Output Signal Handling

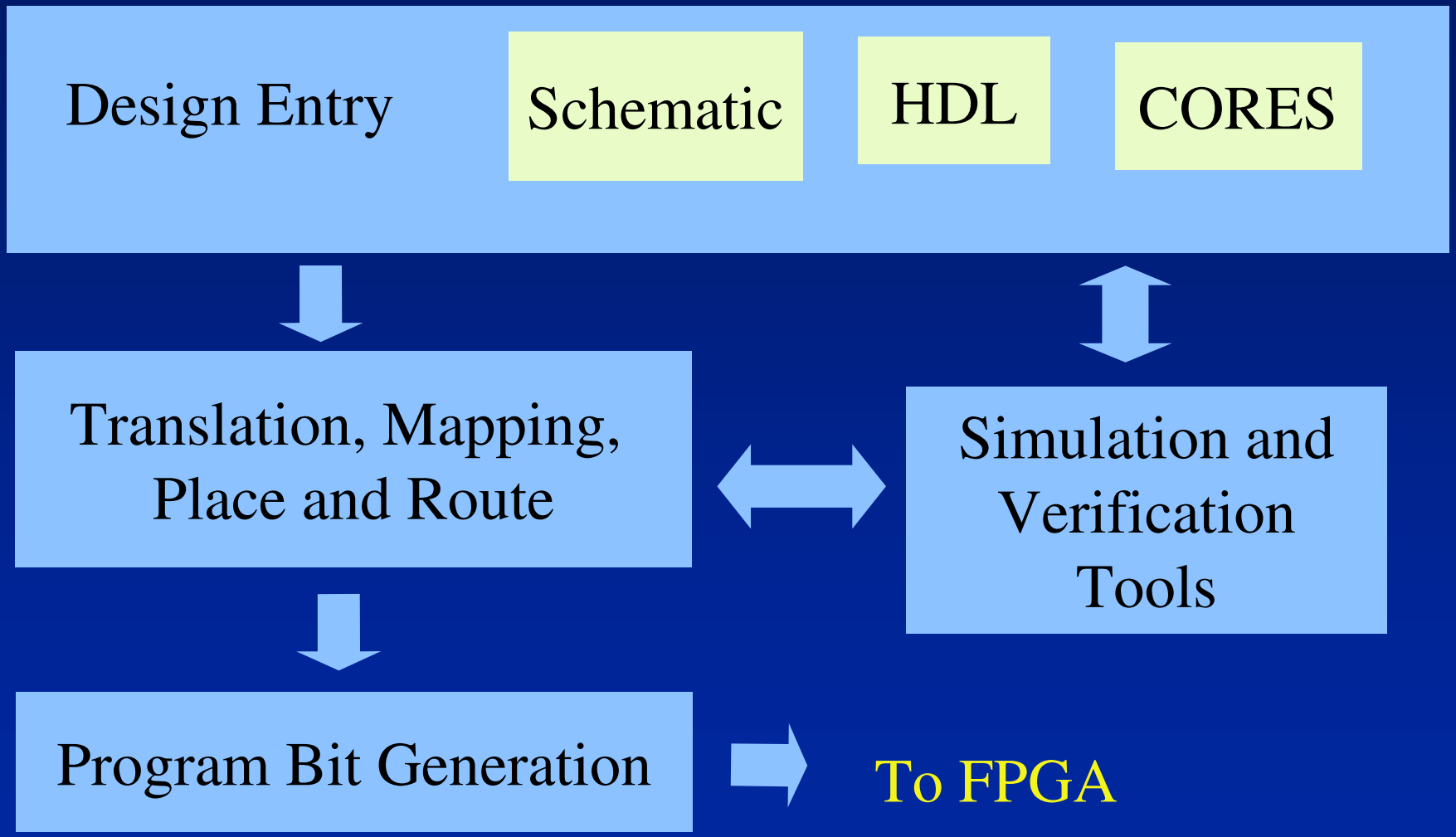
- Implement or Drop In an existing Core for the proper Communication or Bus Interfaces like PCI
- Input Output Ports support wide variety of signaling standards
- Drop In an Asynchronous or Synchronous FIFO
- High Speed Communication is achieved using FPGAs

Virtex FPGAs

Lots of Memory

- Use Memory Compilers to get various Memory Sizes
  - Dual Port Block Memory in the blocks of 4096 memory bits which can be configured as 4096x1, 2048x2, 1024x4, 512x8 and 256x16 memory blocks
  - Distributed Single Port Memories in the units of 16x1 and 32x1
  - Distributed Dual Port Memories in the units of 16x1

Design Flow


Design Entry
Schematic
HDL
CORES
Translation, Mapping, Place and Route
Simulation and Verification Tools
Program Bit Generation
To FPGA

FPGA

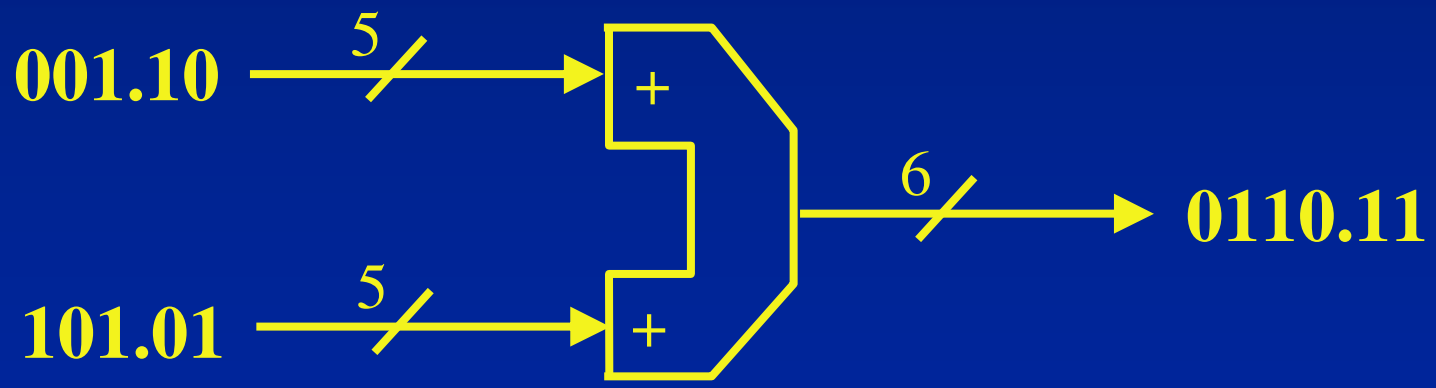
- Hardware in an FPGA is configured by a bit stream generated from the design
  - Reconfigurable Computing is an emerging area
- Current FPGAs can be used if the design fits in the device otherwise choose a larger device

Data Format

- Most Applications store data in Fixed Point Format

0 1 0 1 . 0 1 0 1 = 5.375

- Track Decimal Point properly as we progress along the data path.



Data Format

- Maintain your own precision by discarding appropriate number of bits after each operation
  - Tools like Elanix help do bit true simulation
- No restrictions on the number of bits
  - Use the number of bits required by the application's dynamic range

An example FIR Filter

Simple Code in C

```
y[n] = 0;
for (k = 0; k < N; k++)
    y[n] = y[n] + c[k]*x[n-k];
```

Sequential Commands to the Processor

Efficient Code in C

```
register temp = 0;
for (k = 0; k < N; k++)
    temp = temp + *c_ptr++ * *x_ptr--;
```

An example FIR Filter

Generic VHDL Code

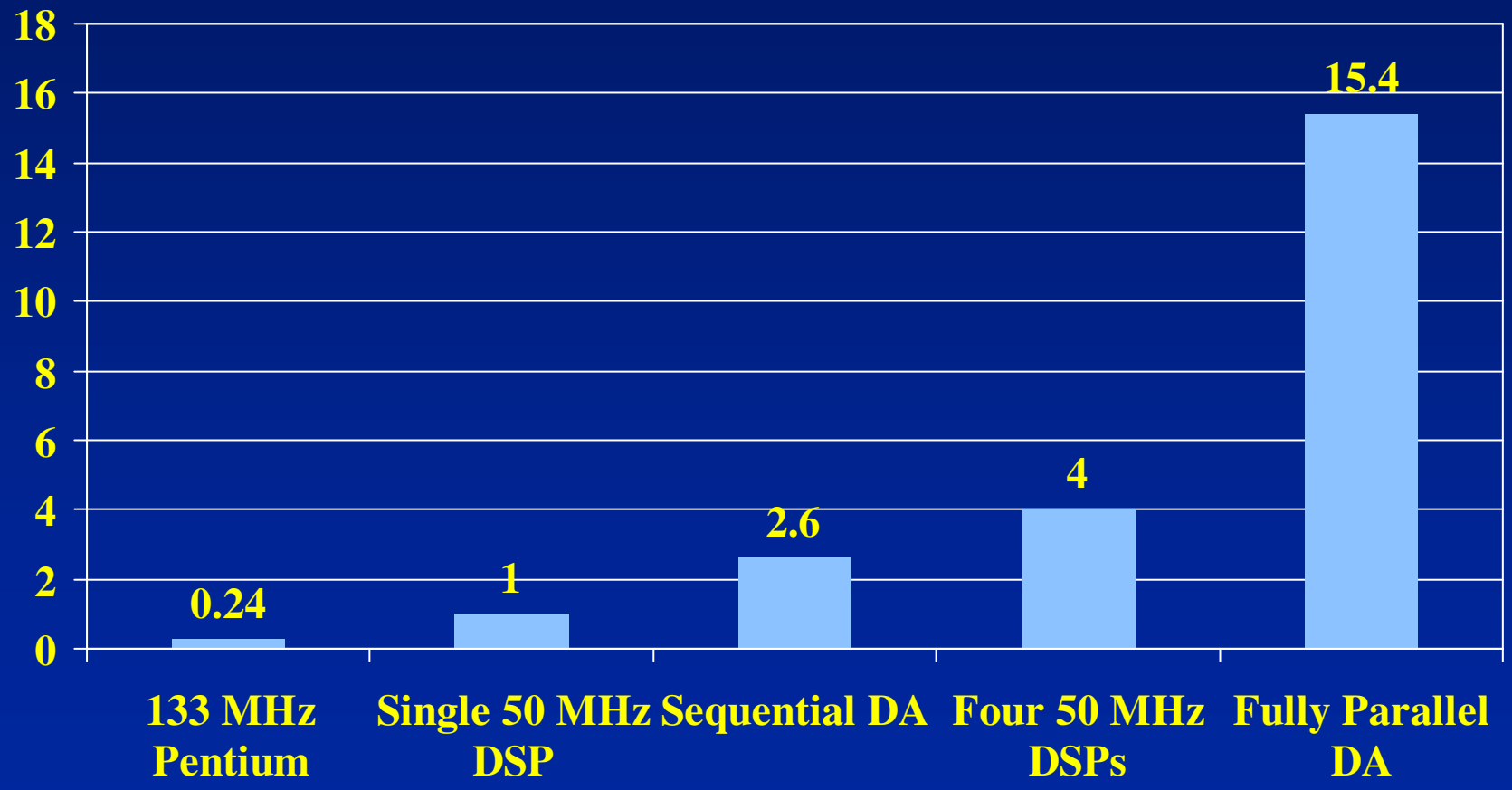
```
.....
mult : FOR k IN 0 TO N-1 GENERATE
        munits : y[n] <= c[k] * x[k];
END GENERATE;
```

Concurrent Statements mapped to Digital Gates in FPGAs

Instantiate Efficient Constant Multiplier

```
.....
mult : FOR k IN 0 TO N-1 GENERATE
        munits : Constant_Coefficient_Multiplier
                GENERIC MAP (const_value => c[k])
                PORT MAP (mult_in => x[k], mult_out => y[n] );
END GENERATE;
.....
```

8-Bit, 16-Tap FIR Filter Performance Comparisons



pDSP vs FPGA

- Use DSP Processor and FPGAs together to achieve high performance DSP solutions
- FPGAs can do compute intensive parts of the DSP System very efficiently
- Use pDSP for control intensive operations
- Use FPGAs and DSP Processor instead of multiple DSP Processors to achieve high speed
- http://www.xilinx.com/products/logicore/dsp/index.htm for more information