Fundación FULGOR

Diseño Digital Avanzado

Unidad 4 - Análisis de Complejidad

Dr. Ariel L. Pola
apola@fundacionfulgor.org.ar
October 11, 2022

Tabla de Contenidos

Introducción

Introducción

Objetivo

- Análisis de complejidad de bloque básicos aplicados en FPGA.

Análisis

- El trabajo de laboratorio consiste en aplicar las diferentes etapas del flujo de diseño sobre tres tipos de sistemas básicos.
- Se analizará la complejidad de implementación de sumadores, multiplicadores y filtros FIR utilizando diferentes técnicas de optimización de complejidad estudiadas en el teórico.
- Los bloque básicos que se analizan son:
  - **Sumadores:** Se compara la complejidad de los sumadores RCA, Hierarchical Carry Select Adder y Binary Carry Look Ahead Adder.
  - **Multiplicadores:** Se compara la complejidad entre un multiplicador convencional y el multiplicador de Booth Modificado.
  - **Filtro FIR:** Se compara la complejidad del FIR aplicando operaciones de producto y sumas convencional y técnicas como Canonic Signed Digit (CSD), Correction Vector (CV), Filtro Directo Transpuesto y Aritmética Distribuida.

Lista de Archivos

Lista de Archivos

Análisis

- A continuación se detallan los archivos que se entregan para el desarrollo del laboratorio.
- Sumadores de 16 bits
  - **Módulos:** rca.v, bcla.v, hierarchicalcsa.v
  - **TestBench:** tb_adder.v
  - **Constraint:** adderConstraint.xdc
- Multiplicadores de 6 bits
  - **Módulos:** mult.v, boothMult.v
  - **TestBench:** tb_mult.v
  - **Constraint:** multConstraint.xdc
- Filtros FIR de 5 coeficientes
  - **Módulos:** FIRfilter.v, FIRfilterCSD.v, FIRfilterCV.v, FIRfilterTDFComp.v, FIRfilterDA.v
  - **TestBench:** tb_FIR_filters.v
  - **Constraint:** firConstraint.xdc

Laboratorio

Paso 1

- Utilizando la herramienta Vivado generar un proyecto para cada escenario de prueba.
- Los escenarios de prueba son:
  - Sumadores
  - Multiplicadores
  - Filtros FIR
- Incluir en el proyecto los archivos entregados por la cátedra para cada escenario.
- Ejemplo: Para el proyecto sumador, en la pestaña **Sources** deberíamos incluir:
  - En la carpeta **Design Sources**: *rca.v*, *bcla.v* y *hierarchicalcsa.v*.
  - En la carpeta **Constraint**: *adderConstraint.xdc*.
  - En la carpeta **Simulation Sources**: *tb_adder.v*.
- Una vez incluido los archivos, ejecutar la simulación comportamental para verificar la correcta instancia de los bloques.

Paso 2

- Elaborar el módulo Top Level que instancie todas las opciones de módulos de cada escenario. El nombre de los puertos de entrada y salida deben ser iguales que los definidos en los archivos **xdc**.
- El diseño debe incluir condicionales de síntesis (‘ifdef - ‘elsif - ‘else - ‘endif) para seleccionar que bloque sintetizar.

Condicional de Compilación

```
1  // `define ADDER_BCLA
2  // `define ADDER_HCSA
3  module topAdder (<ports>);
4     <parameters - variables>
5
6  `ifdef ADDER_BCLA
7     bcla u_bcla (<ports>);
8  `elsif ADDER_HCSA
9     hierarchicalcsa u_hierarchicalcsa (<ports>);
10 `else
11    rca u_rca (<ports>);
12 `endif
13
14 endmodule
```

Paso 3

- Para verificar que el diseño se realizó correctamente, incluir en el TestBench el módulo Top Level y verificar el funcionamiento comparando la salida del módulo Top contra la salida del módulo individual.
- Ejemplo: Para el proyecto sumadores, en la pestaña **Sources** deberíamos incluir
  - El nuevo módulo **TopAdder.v** (diseñado por el alumno) en la carpeta **Design Sources**.
  - En el archivo **tb_adder.v** debemos instanciar el módulo **TopAdder.v** y comparar la salida del módulo **rca.v** del testbench si definimos internamente en el módulo Top el sumador RCA.

Paso 4

- En cada módulo obtener los siguientes parámetros:
  - Síntesis
    - Report Cell Usage
    - Report Instance Areas
    - ROM, RAM, DSP and Shift Register Reporting
  - Esquemáticos
    - Obtener los esquemáticos de **RTL** e **Implementation**.
    - Nota: Generar estos esquemáticos para una sola frecuencia de reloj.
  - Implementación
    - Implementar el diseño variando las frecuencias de reloj para 50MHz, 100MHz y 200MHz.
    - En cada caso obtener el Slack Histogram (Create Slack Histogram) y el **Worst Slack** del camino crítico.

Paso 5

- Elaborar un informe que incluya todos los resultados.
- Analizar cual de las técnicas es menos complejo y cual puede trabajar a mayor frecuencia para cada uno de los escenarios propuestos.