

Práctica 2.2: Técnicas de reducción de consumo en FPGAs

Esta es la continuación de la práctica 2.1. En esta parte se aplicarán diferentes técnicas de reducción de consumo y se realizarán medidas para evaluar su efectividad.

Nota: Para que el circuito funcione, el reset del sistema debe estar desactivado y el PLL debe estar habilitado. Se recuerdan las funcionalidades disponibles controladas mediante los switches y botones (Tabla 1).

Signal	Board Reference	Schematic	Función
PLL_RST	SW1.1	USER_DIP2	Reset PLL: Mantiene el PLL en estado de reset con un '1'.
SYS_RST	S3	USER_PB0	Reset sistema: Mantiene el sistema en estado de reset con un '1'.
CLK_ENA1	SW1.2	USER_DIP1	Enable clkctrl 1: habilita el bloque con un '1'.
CLK_ENA2	SW1.3	USER_DIP0	Enable clkctrl 2: habilita el bloque con un '1'.

Tabla 1: Señales de entrada del diseño y su función.

Parte B: Pipelining

En esta parte deben modificar el bloque multiplicador para variar la cantidad de etapas de pipeline. Para ello, en la ventana *Hierarchy* del Quartus dar doble click sobre cualquiera de los bloques multiplicadores (lpm_mult_32). Los bloques estarán visibles luego de ejecutar *Analysis & Elaboration*. Esto abre el MegaWizard que permite configurar el bloque (no es necesario realizar esto para cada bloque multiplicador instanciado, con modificar uno cambian todos). Ir a la pestaña Pipelining para elegir la latencia del bloque. Activar además la señal de reset (de lo contrario el diseño no compila ya que la instancia de los multiplicadores en el top level contiene dicha señal).

En la parte 1 se usaron multiplicadores puramente combinatorios. Para esta parte deben probar con 1, 2 y 4 etapas de pipeline. Luego de configurar el bloque es necesario descomentar la línea ``define PIPELINE` ubicada al comienzo del archivo `lpmdd_lab2_mult.v`. Este pragma controla si se agregan o no las entradas de reloj y reset de los bloques multiplicadores (ver líneas 73-76 y 88-91 del top-level).

Parte B.1: Estimación de consumo

Para cada caso compilar el diseño, simularlo y estimar el consumo.

Parte B.2: Medidas reales

Para cada caso compilar el diseño, cargarlo en la placa y medir el consumo.

Parte B.3: Bloques hardware

Repetir el experimento configurando los multiplicadores para que utilicen bloques hardware (DSPs) en lugar de lógica. Para ello abrir el Megawizard e ir a General2>Implementation y seleccionar *Use the dedicated multiplier circuit*.

Parte B.4: Frecuencia máxima de operación (opcional)

Además de disminuir el consumo dinámico gracias a la reducción de la propagación de glitches, introducir etapas de pipeline reduce los caminos entre registros. Esto mejora la frecuencia máxima de operación del circuito (aumenta el slack). Verificar esto analizando la frecuencia máxima de operación (Fmax) del circuito al variar las etapas de pipeline.

Parte C: Clock Gating

Una de las ideas más simples para reducir el consumo dinámico de un sistema es apagar temporalmente aquellos bloques o partes del sistema que no estén en uso. Esto se puede lograr con diferentes técnicas¹, por ejemplo, mediante clock gating o clock enable.

El clock gating se define como el bloqueo de la señal de reloj a ciertas partes de un circuito. Se recomienda utilizar solamente los elementos específicos provistos en las FPGAs para este fin y no insertar compuertas comunes en las líneas de reloj. Esto se debe a que insertar compuertas puede afectar la señal de reloj y no necesariamente se apaga toda la red de distribución de reloj dentro del FPGA, cosa que sí se logra utilizando los bloques del fabricante.

La segunda técnica consiste en utilizar flip-flops con entrada clock enable (CE) y deshabilitarla en aquellas partes del circuito que no se está utilizando. De esta forma el flip-flop no puede cambiar su valor. A priori no es tan efectiva como la anterior debido que si bien reduciría el consumo en los flip-flops y compuertas que están deshabilitados, la línea de reloj sigue consumiendo.

Parte C.1: Clock Gating

Para uno de los diseños compilados en la parte anterior, medir el consumo para las siguientes posiciones de los switches 2 y 3: SW[2:3] = "11", "01" y "10". Esto implica:

- "11"- Ambas ramas activas
- "01"- Rama 1 inactiva / Rama 2 activa
- "10"- Rama 1 activa / Rama 2 inactiva

¹ Para más información sobre estas técnicas referirse a <http://arantxa.ii.uam.es/~ivan/spl12-clock-gating.pdf>

Parte D: Opciones de compilación (opcional)

Compilar variando optimizaciones (Performance vs Power vs Área) en:

Settings>Compiler settings>Optimization Mode

Medir consumo para los circuitos sintetizados con las diferentes optimizaciones.

Se deberá entregar un reporte conteniendo lo siguiente:

1. Resultados de todos los experimentos.
2. Cálculo del error en la estimación vs medidas.
3. (Opcional) Comentarios sobre el efecto de las diferentes técnicas y justificación de las diferencias en los resultados de consumo.

Se deberá entregar un único reporte al finalizar las dos partes de esta práctica.



Co-funded by the
Erasmus+ Programme
of the European Union



Disclaimer: The European Commission support for the production of this website does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Anexo 1: Estimación de consumo

Para realizar estimaciones de consumo lo más parecidas a la realidad es necesario seguir los siguientes pasos:

1. Configurar el Quartus para que registre la información de switcheo de todos los nodos durante la simulación en un archivo (formato VCD). Para esto ir a Settings>EDA Tool Settings>Simulation>Options for Power Estimation y seleccionar la opción: *Generate VCD file script*. En *Script Settings...* marcar *All signals*.

En el menú de simulación también se debe configurar el Testbench. Para ello seleccionar la opción *Compile test bench*., luego ir a *Test Benches...>New...* y configurar los siguientes campos:

- Ingresar el nombre del test bench: `lpdd_lab2_mult_tb`
- Ingresar el top level module del test bench: `lpdd_lab2_mult_tb`
- Seleccionar la opción *Use test bench to perform VHDL timing simulation*
- Ingresar el design instance name: DUT
- Seleccionar *End simulation at*: e ingresar el tiempo de simulación: 100us
- En **Test bench and simulation files** agregar el archivo del test bench: `lpdd_lab2_mult_tb.v`

Se debe tener cuidado al elegir el tiempo de simulación, si este es muy corto no alcanzará para capturar el funcionamiento completo del circuito y si es muy largo tomaría demasiado tiempo ya que la simulación Gate-Level es lenta.

2. Luego de compilar, simular el diseño a nivel de compuertas (Gate-Level) para generar el archivo VCD: "Tools>Run Simulation Tool>Gate Level Simulation..."

Esto ejecuta el simulador que se haya configurado en "Tools>Options...>General>EDA Tool Options". En nuestro caso vamos a utilizar ModelSim-Altera. Asegurarse que la ruta del ejecutable del ModelSim está bien configurada en el menú mencionado anteriormente.

3. Por último, para realizar la estimación abrir la herramienta "Processing>Power Analyzer Tool" y realizar las siguientes configuraciones:

Seleccionar "Input file>Use input file to initialize toggle rates and static probabilities during power analysis". Luego ir a "Add Power Input Files..." y seleccionar el archivo VCD generado durante la simulación Gate-Level. Se deben establecer los tiempos iniciales y finales a considerar. El tiempo inicial conviene elegirlo de forma de no considerar el transitorio al comenzar la ejecución y el final debe coincidir con el tiempo de simulación.

Luego ir a "Cooling Solutions and Temperature" e introducir la temperatura ambiente (a efectos de comparar con la medida se debería poner la misma temperatura que había cuando se realizaron las medidas reales). En "Thermal resistance>Use cooling solution" elegir la opción "No heat sink with still air", ya que la placa utilizada no posee ni disipador ni ventilación forzada.

Para estimar presionar Start y luego ver el reporte.

Para ver el resultado de la estimación en el reporte ir a: "Power Analyzer>Current Drawn from Voltage Supplies>Summary". El consumo que interesa (el del core de la FPGA) es el correspondiente a la fila VCC.

Anexo 2: Test para medidas de consumo

La siguiente tabla ayudará a ordenar los resultados de los experimentos. Se da como ejemplo el experimento de medidas de consumo a distintas frecuencias.

Condiciones del experimento

- Temperatura ambiente:
- Tiempo de ejecución previo a medida*:
- R shunt:
- Instrumento medida:
- Configuración del instrumento**:

*- Tiempo necesario para que se establezca la temperatura interna del chip.

** - Base de tiempo, rango de voltaje, puntas, etc.

Medidas reales

Frecuencia (MHz)	Voltaje medido shunt (mV)	Corriente CORE (mA)	Voltaje CORE (V)
30			
40			
50			
Sin Reloj [#]			

[#] - PLL Reset

Estimaciones

Frecuencia (MHz)	Tiempo de simulación (us)	Configuraciones	Corriente CORE (mA)	Voltaje CORE (V)
30		Glitch Filt Power: ON Glitch Filt Power: OFF		
40		Glitch Filt Power: ON Glitch Filt Power: OFF		
50		Glitch Filt Power: ON Glitch Filt Power: OFF		