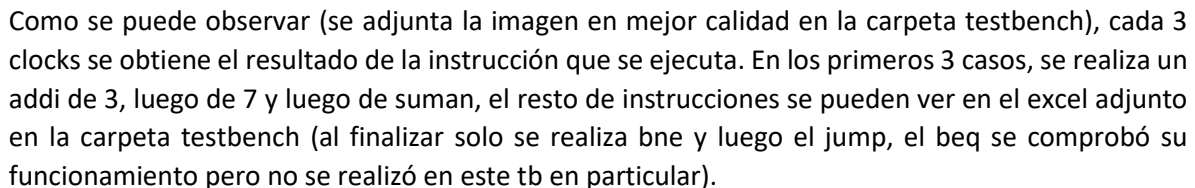


EDUARDO ANAIS

Se realizó la síntesis del mips mediante yosys y los archivos fueron subidos a la carpeta correspondiente. Con un testbench con la memoria fuera del mips se llegó a los siguientes resultados.

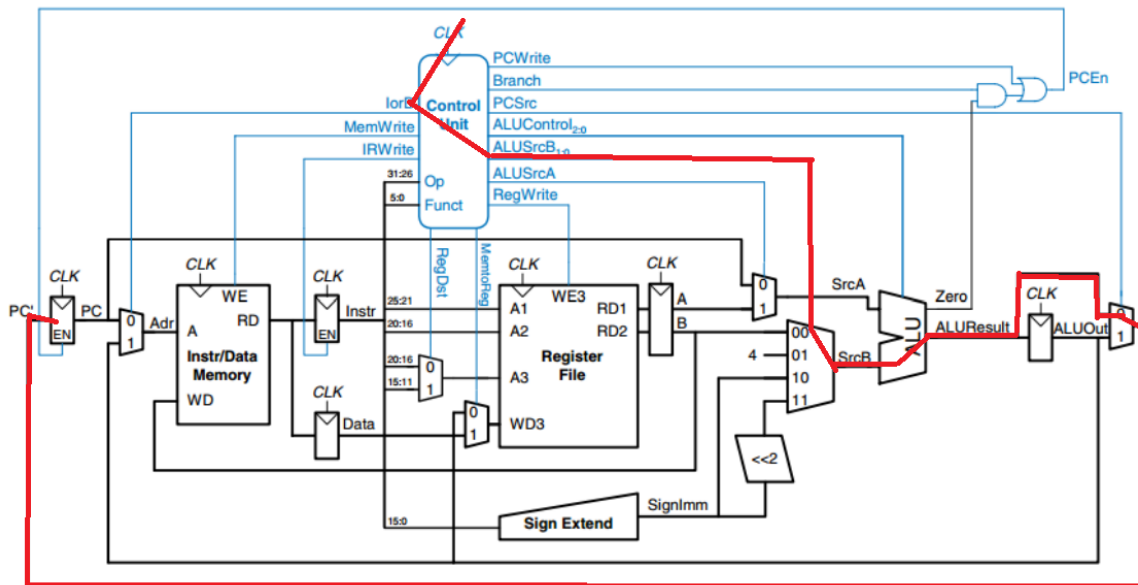


Al reemplazar el mips por el módulo sintetizado con yosys, el resultado del testbench fue el mismo (se puede observar en la carpeta testbench). Además, para corroborar el correcto funcionamiento antes y después de realizar la síntesis, se adjuntan los log con las salidas del monitor, donde se pueden visualizar las mismas señales del testbench pero en texto. La principal diferencia entre ambos archivos, es el comienzo, como al sintetizar se utilizan componentes de acuerdo a una tecnología, en el primer ciclo del clock la salida se encuentra en alta impedancia, esto se debe a que el primer flanco de subido con el reset activado se ejecuta recién en el segundo ciclo y los valores de inicializan en 0.

2. STA

De acuerdo al log generado por el STA, realizado en qflow, el camino o path de mayor retardo corresponde a “Path DFFPOSX1_1220/CLK to DFFPOSX1_1216/D delay 7489.77 ps”. Por lo tanto, se debe respetar este tiempo que corresponde al más lento y “esperarlo”, por decirlo de alguna forma. Al invertir este valor se obtiene la frecuencia máxima del mips **133.511.349 Hz** (133.5 MHz).

El camino más largo tiene sentido ya que incorpora la salida o última etapa del mips que corresponde al cable PC_prima, en este caso es la conexión más próxima a la última etapa y comienza desde el clock del controlador, pasando por el control de estado, lOrD, controlador del multiplexor del ALU source B, la salida del ALU y terminando en el PC_prima. Se presenta un dibujo tentativo del camino más largo.



Efectivamente el diseño cuenta con hold negativo, específicamente, y por lo que se puede apreciar del log del STA, en la memoria del regfile, al parecer corresponde a la memoria 31 (hold at destination = -98.2738). Esto implica que al ocurrir el flanco de subida del clock, el dato (o información) no será el correcto y cambiará después de ejecutarse. Esto sucede por que el delay de la información es mayor al del clock.

Una posible solución es incorporando una etapa al clock (un buffer, por ejemplo,) que aumente su delay y al momento de llegar al regfile el delay del clock sea mayor al del dato.

3. Potencia y Área

NextPCLogic cuenta con un AOI21X1y un INVX1 de acuerdo al log de la síntesis realizado en yosys. Dentro de la librería de la tecnología, las áreas para cada módulo corresponden a 128 y 64 respectivamente, por lo tanto, el **área** total es de 192 unidades de área que, al parecer, no se encuentran especificadas en la cabecera de la librería.

La **potencia estática** se obtiene del leakage power de cada celda 0.034284 nW + 0.0152465 nW = 0.0495305 nW respectivamente.

Para calcular la **potencia dinámica** se utiliza la fórmula vista en clases y de acuerdo al factor de actividad de cada nodo se calcula, las capacitancias se obtienen de la librería. De la A a la C corresponden a las entradas del AOI y la C_I a la capacitancia del inversor y la capacitancia load corresponde a la del BUFX2 (enable de un registro se conecta por medio de un BUFX2 de acuerdo a lo visto en el archivo sintetizado).

Suponiendo que la probabilidad de las entradas es de 0.5 se realizó el cálculo de acuerdo a la tabla vista en la tarea 4.

*Potencia*_{dinámica}

$$= C_A * \frac{1}{4} * V_{dd}^2 * f + C_B * \frac{1}{4} * V_{dd}^2 * f + C_C * \frac{1}{4} * V_{dd}^2 * f + C_I * \frac{15}{64} * V_{dd}^2 * f \\ + C_{load} * \frac{15}{64} * V_{dd}^2 * f$$

C_A = 0.0265613 pF

C_B = 0.0268633 pF

C_C = 0.02277 pF

C_I = 0.0134094 pF

C_load = 0.0134147 pF