



Sesiones FPGA con EDK. Preguntas frecuentes

Hipólito Guzmán Miranda
Departamento de Ingeniería Electrónica
Universidad de Sevilla
hguzman@us.es

Preguntas Frecuentes:

[Me quedo sin memoria muy rápido. ¿Cómo puedo incrementar la memoria de instrucciones o la de datos?](#)

En System Assembly View puedes cambiar el "Size" de `i1mb_cntrl` y de `d1mb_cntrl`. Por supuesto el tamaño máximo está limitado por la capacidad de la FPGA.

Es posible que tengas que regenerar el script de linkado para que el compilador se entere de que has cambiado el tamaño de la memoria. En Software -> Generate Linker Script puedes generar un nuevo script de linkado con los tamaños de la memoria correctos (te aparecerá un asistente con los nuevos valores automáticamente, y puedes aceptar los valores por defecto).

[Me sigo quedando sin memoria, por mucho que incremente los tamaños de las memorias internas.](#)

Probablemente estás haciendo llamadas a la función `printf()`. Imprimir con formato es una función muy costosa en ocupación de memoria, tanto en instrucciones como en uso de pila (stack) y montón (heap).

Según Xilinx tenemos disponibles las siguientes funciones para imprimir texto:

Function	Size	Limitations
<code>printf()</code>	51788	None, full featured
<code>iprintf()</code>	18294	No floating point, reentrant
<code>xil_printf()</code>	2953	No floating point, not reentrant(single thread only), no longlong(64 bit)
<code>putnum()</code>	284	Integer to HEX only, no other formats
<code>print()</code>	185	No numbers output, just strings

"All of these functions can be prototyped by including `<stdio.h>`."

Fíjate como `printf()` ocupa 51 Kbytes! Para realizar estas prácticas podemos utilizar únicamente `print()` y `putnum()`, que son las que menos memoria ocupan.

[Estoy usando la función `selftest` que he encontrado en los drivers generados por EDK y me da montones de errores de compilación](#)

Recuerda que debes cambiar las llamadas a `xil_io_out32` y `xil_io_in32` como se explica más arriba.



Adicionalmente, cuidado con el uso de `xil_printf()`, ya que la función de `selftest` la usa bastante. Puede ser más útil utilizar únicamente `print()` y `putnum()`, o incluso no imprimir nada y simplemente procesar correctamente el valor devuelto por `selftest` en el `main`.

¿Cómo puedo incrementar el tamaño de la pila (stack)?

Si estás viendo este error al compilar el software:

```
region ilmb_cntlr_dlmb_cntlr is full (TestApp_Memory_microblaze_0/executable.elf  
section .stack)
```

y ya has eliminado todos los `print*` innecesarios, te puede ser interesante incrementar el tamaño de la pila. Para ello hay que regenerar el script de linkado de la aplicación software.

Para ésto, lo primero que deberías hacer es un backup del fichero, que está en
<carpeta_del_proyecto>/TestApp_Memory_microblaze_0/src/TestApp_Memory_LinkScr.ld

Tras eso, en Software -> Generate Linker Script puedes cambiar el tamaño asignado a la pila, de 0x400 a por ejemplo 0x800.

Advertencia: Un error en el script de linkado puede hacer que el microprocesador deje de funcionar.

¿Puedo hacer `sleep/usleep` en MicroBlaze?

En la configuración por defecto del microprocesador, realmente no, ya que en dicha configuración MicroBlaze no tiene timers internos. Pero podemos implementar un delay con el siguiente bucle `for`:

```
for (j = 0; j < DELAY; j++) { asm("nop"); }
```