

## ¿Cuánto tengo de memoria RAM/ROM distribuida/bloques en mi tarjeta?

Los procesos escritos a continuación, son aplicables en la creación de módulos de RAM, ROM, distribuida o de bloques.

Dependiendo de la FPGA, se tiene disponible cierta cantidad de memoria distribuida y otra destinada para memoria de bloques. Es importante tener en cuenta que la memoria distribuida utiliza los LUTs de la FPGA. Por lo tanto, es conveniente tomar en cuenta que, si el diseño requiere de una cantidad pequeña de memoria, la implementación puede llevarse a cabo en VHDL o RAM distribuida. Si el sistema requiere de mayor cantidad de memoria, debemos utilizar los bloques de RAM de la FPGA. Para memorias más grandes, podemos acceder a la memoria RAM o FLASH (externas a la FPGA) de la tarjeta.

A continuación, la tabla muestra las cantidades de memoria configurable en la FPGA.

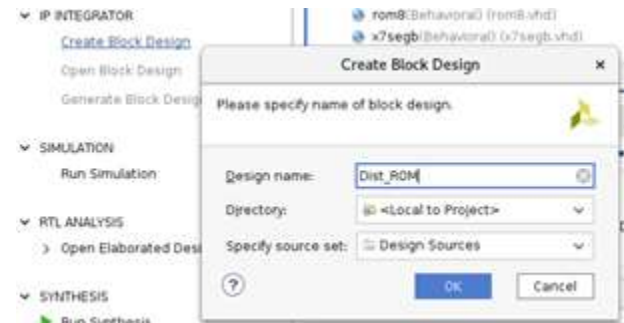
Table 4: Artix-7 FPGA Feature Summary by Device

Device	Logic Cells	Configurable Logic Blocks (CLBs)		DSP48E1 Slices <sup>(2)</sup>	Block RAM Blocks <sup>(3)</sup>			CMTs <sup>(4)</sup>	PCIe <sup>(5)</sup>	GTPs	XADC Blocks	Total I/O Banks <sup>(6)</sup>	Max User I/O <sup>(7)</sup>
		Slices <sup>(1)</sup>	Max Distributed RAM (Kb)		18 Kb	36 Kb	Max (Kb)						
XC7A12T	12,800	2,000	171	40	40	20	720	3	1	2	1	3	150
XC7A15T	16,640	2,600	200	45	50	25	900	5	1	4	1	5	250
XC7A25T	23,360	3,650	313	80	90	45	1,620	3	1	4	1	3	150
XC7A35T	33,280	5,200	400	90	100	50	1,800	5	1	4	1	5	250
XC7A50T	52,160	8,150	600	120	150	75	2,700	5	1	4	1	5	250
XC7A75T	75,520	11,800	892	180	210	105	3,780	6	1	8	1	6	300
XC7A100T	101,440	15,850	1,188	240	270	135	4,860	6	1	8	1	6	300

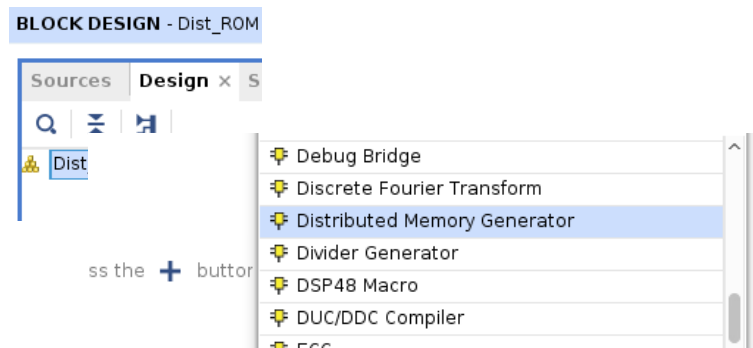
\*\*\*La tarjeta NEXYS4 DDR tiene la FPGA XC7A100T y la BASYS 3 tiene XC7A35T.

## RAM/ROM distribuida/bloques en VIVADO

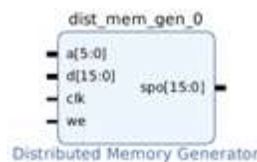
- En la opción IP INTEGRATOR → Create Block Design:
- Proporcionamos el nombre del bloque a crear.



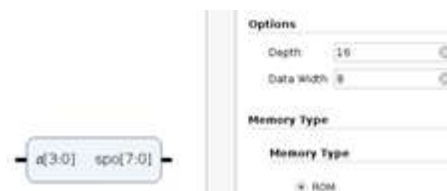
- Aparece una ventana para proporcionar las características del bloque.
- Al presionar el signo “+”, aparece otra ventana para seleccionar el tipo de bloque.



- En la figura que aparece, damos “doble click” para proporcionar las características.



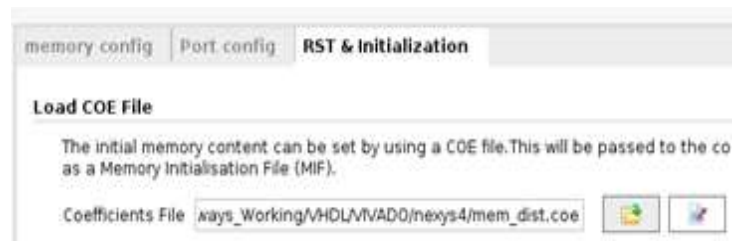
- Proporcionamos el tipo y el tamaño:



Configuración de los puertos:



- Y, si vamos a cargar un archivo coe (coeficient file), lo definimos en este momento. El archivo se puede escribir con cualquier editor de texto o en esta misma ventana oprimiendo el ícono



- Aquí el ejemplo de uno que creamos:

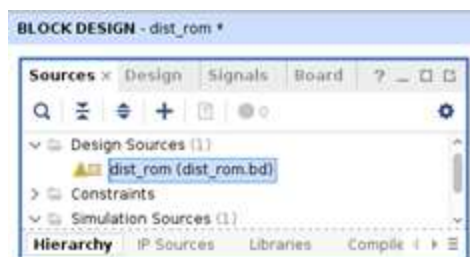
```
; Example initialization file for a 16x8 distributed ROM
;
```

```
memory_initialization_radix = 16;
memory_initialization_vector =
0 C8 F9 AF
64 95 6C D4
39 E7 5A 96
84 37 28 4C;
```

- Damos “OK” y termina la edición.
- Posteriormente podemos editar nuevamente las características del módulo:
  - Nos ubicamos en la entrada o salida, damos botón derecho y hacemos las señales externas con la opción *botón derecho + Make External*.
  - Nos ubicamos en la la entrada o salida, damos *botón derecho + External Port Properties*.
    - Cambiamos el nombre y damos “enter”. De tal manera que nuestro bloque puede quedar como:



- Para incluir el bloque en nuestro diseño “top level”, hacemos lo siguiente:
  - Seleccionamos el bloque:



- Finalmente, damos *botón derecho + Create HDL Wrapper + Copy generated wrapper + ok*:
  - En nuestro diseño “top level” incluimos la instancia:
 

```
Inst_dist_rom: entity work.dist_rom_wrapper PORT MAP(
    addr => addr,
    M => M
);
```