

Cómo crear un fichero COE

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2024/11/05/como-crear-un-fichero-coe/>

Blog: <https://soceame.wordpress.com/>

GitHub: <https://github.com/DRubioG>

Fecha última modificación: 23/02/2025

(Esto es un extracto de una entrada anterior donde explico cómo crear un fichero de coeficientes COE)

<https://soceame.wordpress.com/2024/11/05/como-implementar-memorias-en-vivado-parte-2-memoria-rom/>

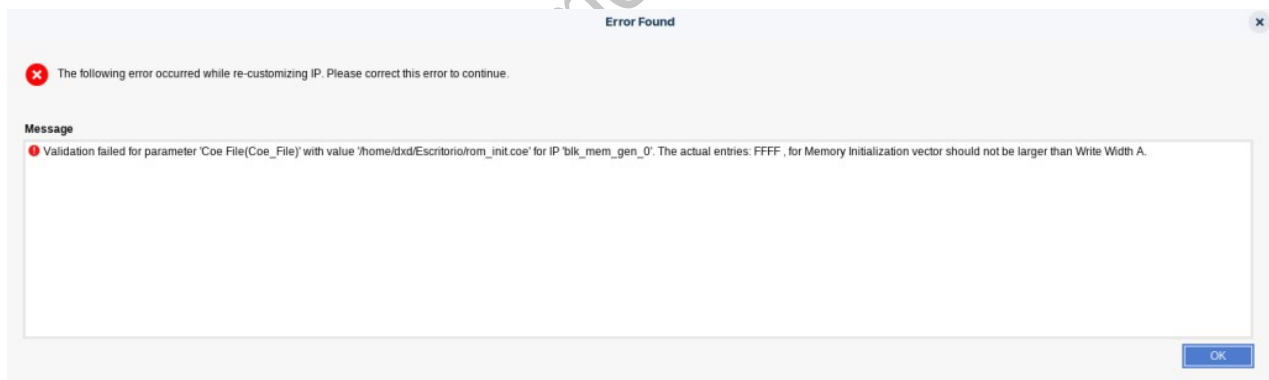
Un fichero COE es un tipo de archivo que contiene información para grabarla en una ROM de Xilinx.

Este tipo de ficheros se basa en dos filas, una que indica en que formato están guardados los datos, **MEMORY_INITIALIZATION_RADIX**, y otra fila que contiene exactamente estos datos en formato matriz, **MEMORY_INITIALIZATION_VECTOR**.

NOTA: es importante aclarar cosas aquí.

El **MEMORY_INITIALIZATION_RADIX** nos dice en que formato están representados los datos (16: hexadecimal, 8: octal, 2: binario, etc), lo que **NO nos dice es el tamaño de los datos que se van a utilizar** (si son de 16 bits, 13 bits, 10 bits, etc), esto último **NO** se define en ningún sitio. Entonces, los coeficientes se ajustan al tamaño de la salida.

Para que lo entendamos mejor. Yo puedo tener un **MEMORY_INITIALIZATION_RADIX** de 16 (formato hexadecimal) y tener un tamaño de datos de 10bits. Entonces, lo que tiene que ocurrir es que ningún coeficiente en formato hexadecimal puede exceder los 10 bits (entonces, los datos solo pueden ir del 00 al 3FF). Si no se produce un error al cargar los datos.



Esto es un pequeño ejemplo de fichero:

```
MEMORY_INITIALIZATION_RADIX=16;
MEMORY_INITIALIZATION_VECTOR=
11,
22,
33,
44,
55,
66,
77,
88,
99,
aa,
```

```
bb,  
cc,  
dd,  
ee,  
ff,  
00,  
a1,  
a2,  
a3,  
a4,  
a5,  
a6,  
a7,  
a8,  
b1,  
b2,  
b3,  
b4,  
b5,  
b6,  
b7,  
b8;
```

En la primera parte se puede ver que se le dice que los datos están en representación hexadecimal, y en la segunda parte se guardan los datos.

Entonces, para que este ejemplo pueda ser utilizado como coeficientes en una ROM, **la salida tiene que ser como mínimo de 8 bits** (porque uno de los coeficientes es 'ff')

NOTA: es recomendable que la profundidad de la memoria se igual que la de los coeficientes para evitar que sobre memoria (que estará llena de ceros)