

Configurar los periféricos de un PolarFire SoC

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2025/01/07/configurar-los-perifericos-de-un-polarfire-soc/>

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

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

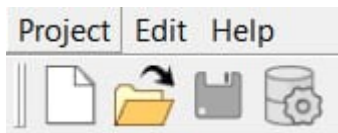
Fecha última modificación: 24/02/2025

Para activar los periféricos internos del SoC de un PolarFire SoC hay que abrir el programa **PFSoc MSS Configurator**. Este es el programa que configura el SoC que luego se tiene que importar a Libero como un *MSS SmartDesign*.



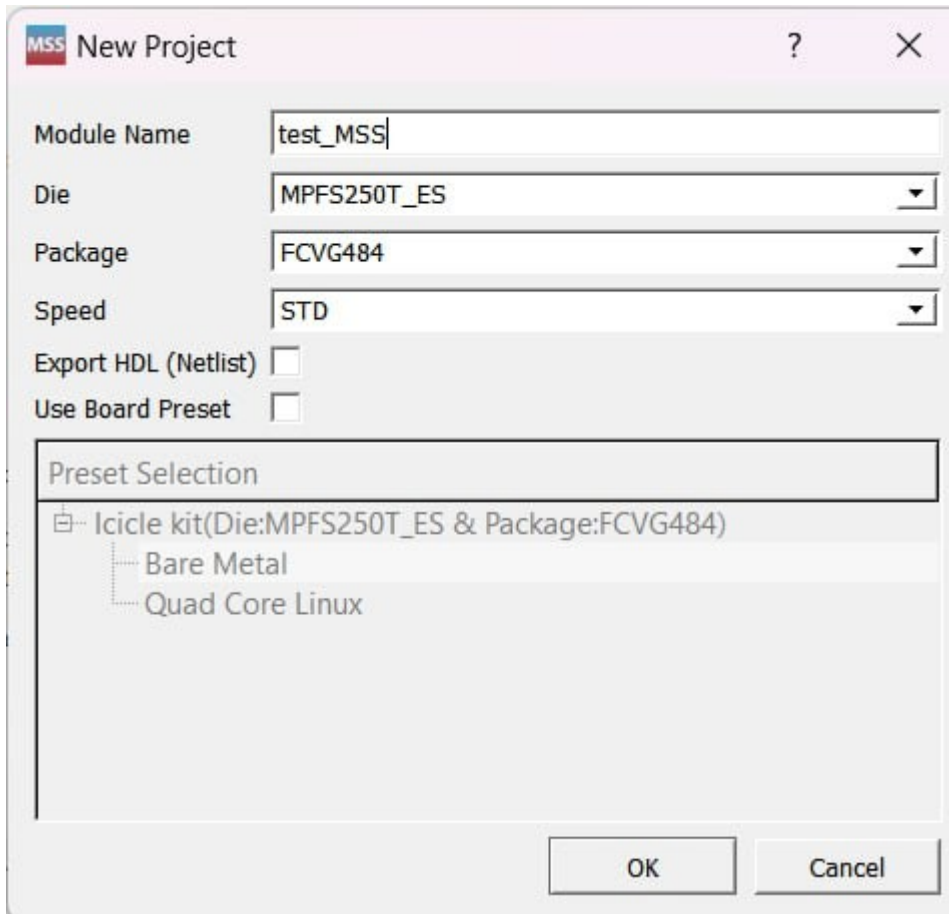
PFSoc MSS Configurator

Al abrir el programa nos dan dos opciones, crear una nueva configuración o abrir una existente.

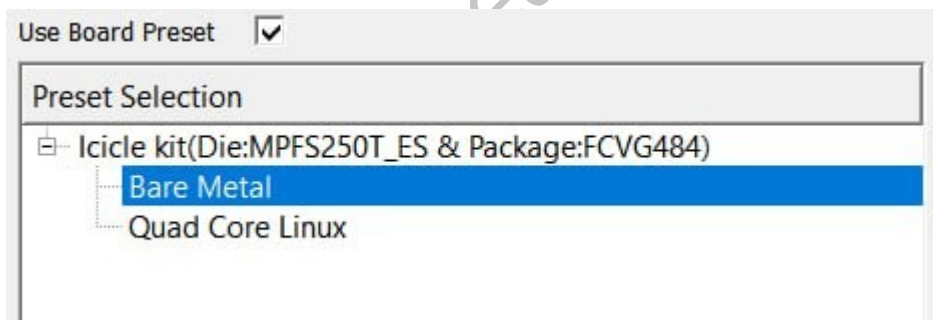


- **Crear una nueva configuración**

Lo primero que hace es abrirnos una pestaña dónde nos va a preguntar sobre que tipo de PolarFire vamos a configurar.



También tiene un opción que nos permite usar una placa que ya esté preconfigurada en el programa. De momento, solo hay configurada la Icicle Kit.



- **Abrir una existente**

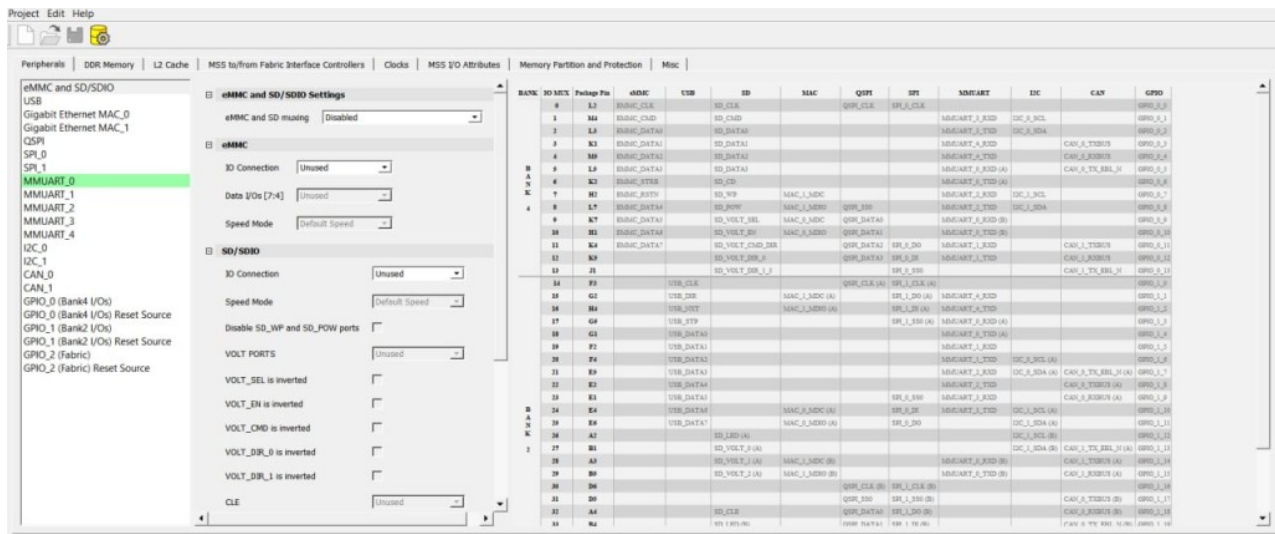
Para abrir una existente nos pedirá que le indiquemos el fichero .cfg con la configuración. Desde este fichero de configuración nos abrirá la configuración previamente guardada.

Aquí tienes un proyecto base para el PolarFire SoC Discovery Kit:

https://github.com/DRubioG/Polarfire_basic_project/tree/main/PolarFire_config

NOTA: lo más recomendable es trabajar sobre una configuración existente debido a que una de las características que pide para configurar el SoC es la memoria RAM que tiene el SoC.

Una vez se abre el programa se puede ver una pestaña como la siguiente, en esta pestaña se pueden ver todas las posibles configuraciones de la PolarFire SoC.



Este programa permite configurar las interfaces para diferentes periféricos (*Peripherals*). Por ejemplo, permite hasta 5 UARTs, 2 I2Cs, 2 SPIs, GPIOs por los bancos existentes.



Los pines específicos del SoC son los pines llamados «**MSS I/Os Bank ...**», estos pines están multiplexados, por lo que no es posible utilizar todas las interfaces del SoC al mismo tiempo por estos pines. Para solucionar esto aparecen los pines llamados **Fabric I/O**, que son pines que no pertenecen al SoC, si no a la lógica programable de la FPGA. Entonces, es el usuario el que elige los pines de salida de la interfaz después de sintetizar.

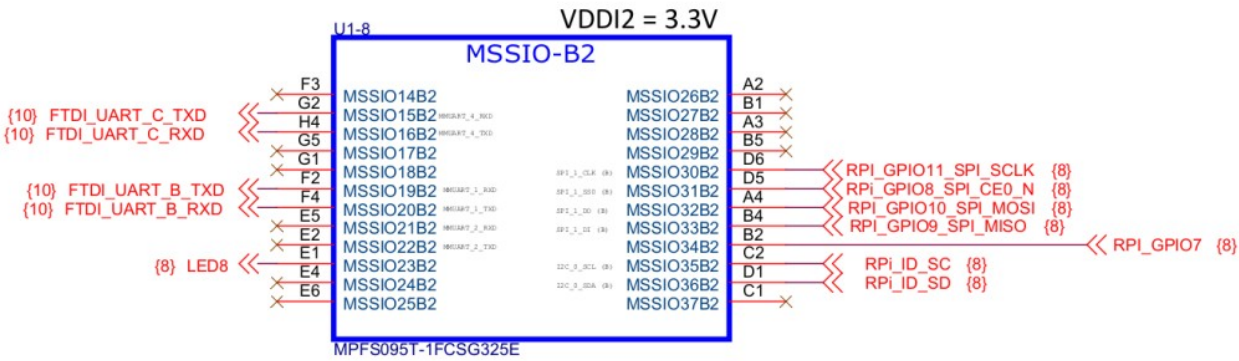
The screenshot shows a configuration window for the MMUART_0 peripheral. On the left, a list of peripherals is visible, with MMUART_0 selected. The main configuration area for MMUART_0 includes fields for IO Connection, Mode, Modem Signals, and Other. A dropdown menu is open for the IO Connection field, displaying a list of options: Unused, MSS I/Os Bank2 A, MSS I/Os Bank2 B, MSS I/Os Bank2 C, MSS I/Os Bank4 A, MSS I/Os Bank4 B, and Fabric I/O. The Fabric I/O option is currently selected and highlighted in blue.

La tabla de pines multiplexados es la siguiente, donde se puede ver que los pines 17 y 18 están seleccionados. Estos pines están en el banco 2 del SoC.

NOTA: en esta tabla de pines figura el nombre específico del periférico en los pines, por lo que se vuelve más fácil saber que pines configurar.

BANK	IO MUX	Package Pin	EMMC	USB	SD	MAC	QSPI	SPI	MMUART	I2C	CAN	GPIO
BANK 4	0	L2	EMMC_CLK		SD_CLK		QSPI_CLK	SPI_0_CLK				GPIO_0_0
	1	M4	EMMC_CMD		SD_CMD				MMUART_3_RXD	I2C_0_SCL		GPIO_0_1
	2	L3	EMMC_DATA0		SD_DATA0				MMUART_3_TXD	I2C_0_SDA		GPIO_0_2
	3	K1	EMMC_DATA1		SD_DATA1				MMUART_4_RXD		CAN_0_TXBUS	GPIO_0_3
	4	M5	EMMC_DATA2		SD_DATA2				MMUART_4_TXD		CAN_0_RXBUS	GPIO_0_4
	5	L5	EMMC_DATA3		SD_DATA3				MMUART_0_RXD (A)		CAN_0_TX_EBL_N	GPIO_0_5
	6	K2	EMMC_STRB		SD_CD				MMUART_0_TXD (A)			GPIO_0_6
	7	H2	EMMC_RSTN		SD_WP	MAC_1_MDC			MMUART_2_RXD	I2C_1_SCL		GPIO_0_7
	8	L7	EMMC_DATA4		SD_POW	MAC_1_MDIO	QSPI_SS0		MMUART_2_TXD	I2C_1_SDA		GPIO_0_8
	9	K7	EMMC_DATA5		SD_VOLT_SEL	MAC_0_MDC	QSPI_DATA0		MMUART_0_RXD (B)			GPIO_0_9
	10	H1	EMMC_DATA6		SD_VOLT_EN	MAC_0_MDIO	QSPI_DATA1		MMUART_0_TXD (B)			GPIO_0_10
	11	K4	EMMC_DATA7		SD_VOLT_CMD_DIR		QSPI_DATA2	SPI_0_DO	MMUART_1_RXD		CAN_1_TXBUS	GPIO_0_11
	12	K5			SD_VOLT_DIR_0		QSPI_DATA3	SPI_0_DI	MMUART_1_TXD		CAN_1_RXBUS	GPIO_0_12
BANK 2	13	J1			SD_VOLT_DIR_1_3			SPI_0_SS0			CAN_1_TX_EBL_N	GPIO_0_13
	14	F3		USB_CLK			QSPI_CLK (A)	SPI_1_CLK (A)				GPIO_1_0
	15	G2		USB_DIR		MAC_1_MDC (A)		SPI_1_DO (A)	MMUART_4_RXD			GPIO_1_1
	16	H4		USB_NXT		MAC_1_MDIO (A)		SPI_1_DI (A)	MMUART_4_TXD			GPIO_1_2
	17	G5		USB_STP				SPI_1_SS0 (A)	MMUART_0_RXD (A)			GPIO_1_3
	18	G1		USB_DATA0					MMUART_0_TXD (A)			GPIO_1_4
	19	F2		USB_DATA1					MMUART_1_RXD			GPIO_1_5
	20	F4		USB_DATA2					MMUART_1_TXD	I2C_0_SCL (A)		GPIO_1_6
	21	E5		USB_DATA3					MMUART_2_RXD	I2C_0_SDA (A)	CAN_0_TX_EBL_N (A)	GPIO_1_7
	22	E2		USB_DATA4					MMUART_2_TXD		CAN_0_TXBUS (A)	GPIO_1_8
	23	E1		USB_DATA5				SPI_0_SS0	MMUART_3_RXD		CAN_0_RXBUS (A)	GPIO_1_9
	24	E4		USB_DATA6		MAC_0_MDC (A)		SPI_0_DI	MMUART_3_TXD	I2C_1_SCL (A)		GPIO_1_10
	25	E6		USB_DATA7		MAC_0_MDIO (A)		SPI_0_DO		I2C_1_SDA (A)		GPIO_1_11
	26	A2			SD_LED (A)					I2C_1_SCL (B)		GPIO_1_12
	27	B1			SD_VOLT_0 (A)					I2C_1_SDA (B)	CAN_1_TX_EBL_N (A)	GPIO_1_13
	28	A3			SD_VOLT_1 (A)	MAC_1_MDC (B)			MMUART_0_RXD (B)		CAN_1_TXBUS (A)	GPIO_1_14
	29	B5			SD_VOLT_2 (A)	MAC_1_MDIO (B)			MMUART_0_TXD (B)		CAN_1_RXBUS (A)	GPIO_1_15
	30	D6					QSPI_CLK (B)	SPI_1_CLK (B)				GPIO_1_16
	31	D5					QSPI_SS0	SPI_1_SS0 (B)			CAN_0_TXBUS (B)	GPIO_1_17
	32	A4			SD_CLE		QSPI_DATA0	SPI_1_DO (B)			CAN_0_RXBUS (B)	GPIO_1_18
	33	B4			SD_LED (B)		QSPI_DATA1	SPI_1_DI (B)			CAN_0_TX_EBL_N (B)	GPIO_1_19
	34	B2			SD_VOLT_0 (B)		QSPI_DATA2				CAN_1_TXBUS (B)	GPIO_1_20
	35	C2			SD_VOLT_1 (B)	MAC_0_MDC (B)	QSPI_DATA3		MMUART_0_RXD (C)	I2C_0_SCL (B)	CAN_1_RXBUS (B)	GPIO_1_21
	36	D1			SD_VOLT_2 (B)	MAC_0_MDIO (B)			MMUART_0_TXD (C)	I2C_0_SDA (B)	CAN_1_TX_EBL_N (B)	GPIO_1_22
	37	C1					QSPI_CLK (C)	SPI_0_CLK				GPIO_1_23

Estos pines se corresponden con las conexiones establecidas en el esquemático de la placa para el banco 2 del SoC.



Como se puede ver, el I2C también tiene la opción de seleccionar el tipo de pin por el que se puede establecer la conexión.

I2C_0

IO Connection	Unused
Speed Mode	Unused
Baud Rate Clock Source	MSS I/Os Bank2 A
System Management Bus Signals	Unused

Con los GPIO igual.

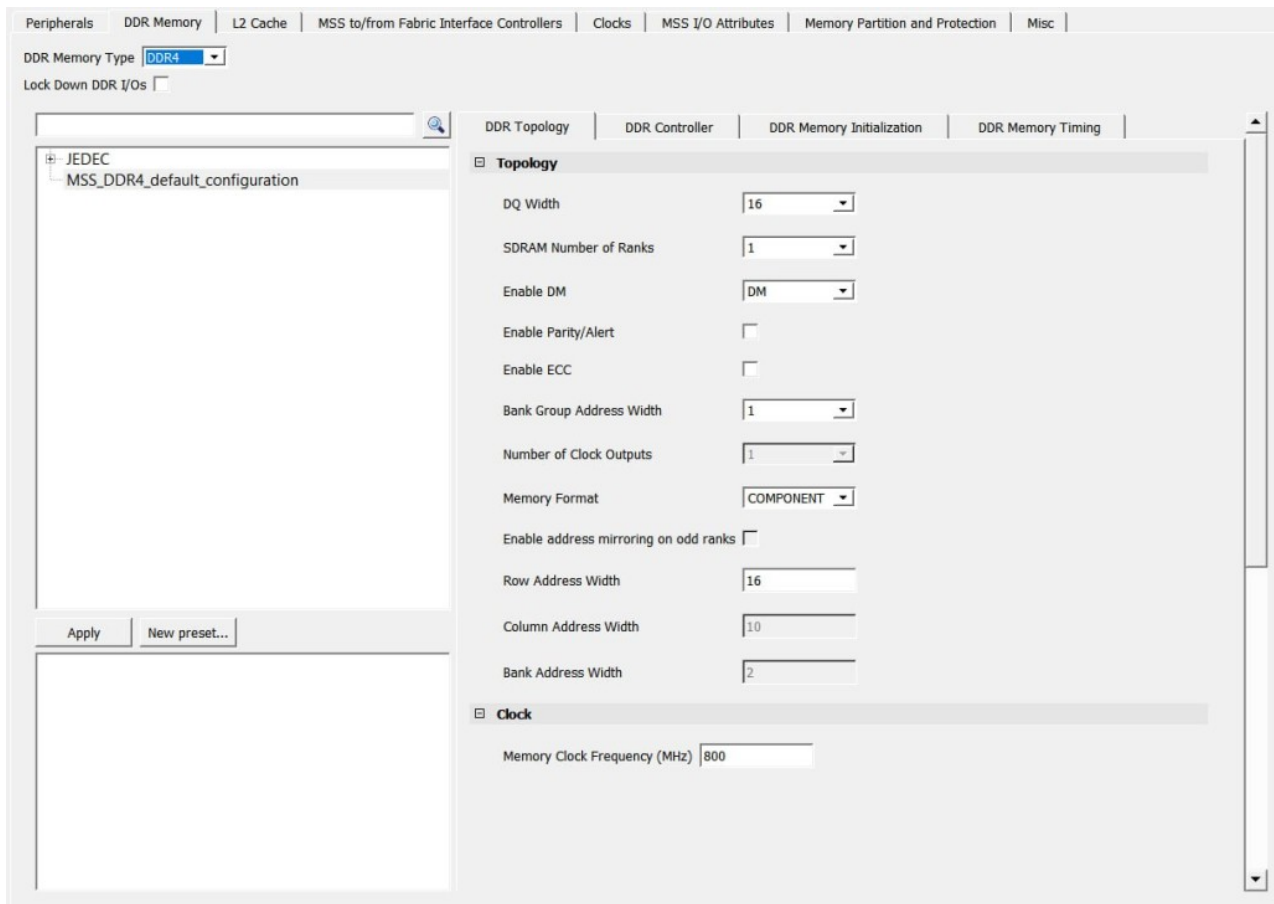
GPIO_1 (Bank2 I/Os)

GPIO_1_0	Unused
GPIO_1_1	MSS I/Os Bank2

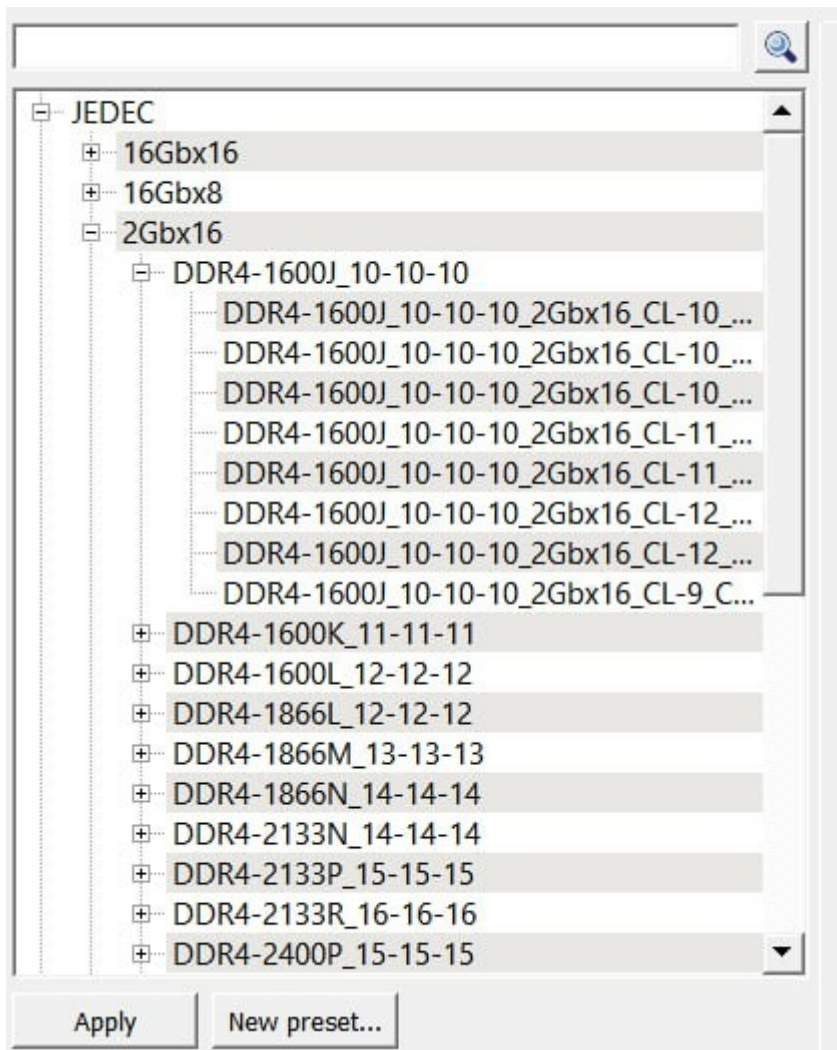
Ahora si se mira los GPIOs, aparecen dos tipos de GPIOs, los que van por los pines del SoC y los que solo pueden ir por la lógica programable (*Fabric*).

eMMC and SD/SDIO	GPIO_1 (Bank2 I/Os)			
USB	GPIO_1_0	Unused	Direction	Input
Gigabit Ethernet MAC_0	GPIO_1_1	Unused	Direction	Input
Gigabit Ethernet MAC_1	GPIO_1_2	Unused	Direction	Input
QSPI	GPIO_1_3	Unused	Direction	Input
SPI_0	GPIO_1_4	Unused	Direction	Input
SPI_1	GPIO_1_5	Unused	Direction	Input
MMUART_0	GPIO_1_6	Unused	Direction	Input
MMUART_1	GPIO_1_7	Unused	Direction	Input
MMUART_2	GPIO_1_8	Unused	Direction	Input
MMUART_3	GPIO_1_9	Unused	Direction	Input
MMUART_4	GPIO_1_10	Unused	Direction	Input
I2C_0	GPIO_1_11	Unused	Direction	Input
I2C_1				
CAN_0				
CAN_1				
GPIO_0 (Bank4 I/Os)				
GPIO_0 (Bank4 I/Os) Reset Source				
GPIO_1 (Bank2 I/Os)				
GPIO_1 (Bank2 I/Os) Reset Source				
GPIO_2 (Fabric)				
GPIO_2 (Fabric) Reset Source				

La siguiente pestaña nos permite configurar la memoria RAM.



Tiene un motón de memorias RAM dentro, por ello, salvo que se sepa específicamente que memoria se tiene en el SoC es mejor que tengas un perfil preconfigurado.



La siguiente pestaña permite configurar la L2 Cache.

L2-LIM WAYS (128 KB for each WAY)

Scratchpad WAYS (128 KB for each WAY)

Initiators	L2 Cache Size	WAY0	WAY1	WAY2	WAY3	WAY4	WAY5	WAY6	WAY7	WAY8	WAY9	WAY10	WAY11	WAY12	WAY13
DMA	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓	Scratchpad 512 KB					
FPGA Port0	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
FPGA Port1	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
FPGA Port2	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
FPGA Port3	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
E51 D\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
E51 I\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
U54_1 D\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
U54_1 I\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
U54_2 D\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
U54_2 I\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
U54_3 D\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
U54_3 I\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
U54_4 D\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						
U54_4 I\$	1024 KB	✓	✓	✓	✓	✓	✓	✓	✓						

Note1: 'L2 Cache Size' in each row is the size accessible by that particular Initiator out of 'Total Shared L2 Cache Size' (1024 KB)

Note2: Initiator cannot evict from cache when WAY is unselected

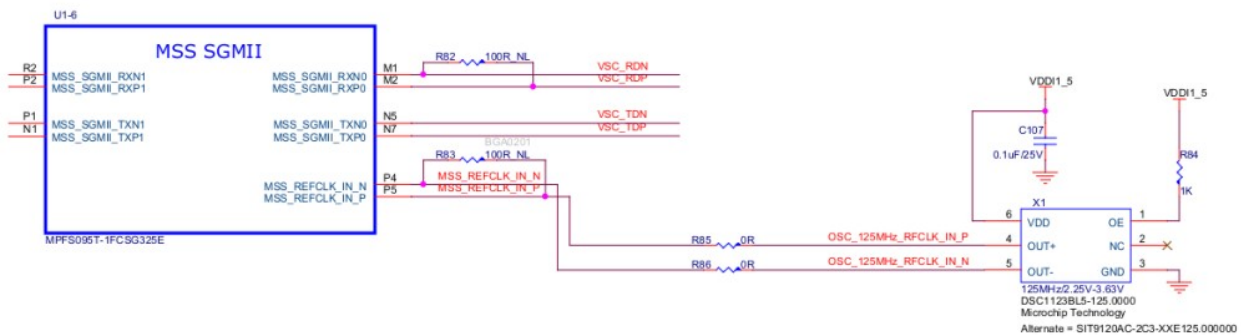
La siguiente pestaña es importante, debido a que es la que configura la interfaz de comunicación con los diferentes bloques IP que se utilicen en el SoC. Se puede ver que hay hasta 4 perfiles, 3 para AXI y uno para APB.

Peripherals	DDR Memory	L2 Cache	MSS to/from Fabric Interface Controllers	Clocks	MSS I/O
FIC_0 (AXI4)					
Use Initiator Interface	<input type="checkbox"/>	Use Target Interface	<input type="checkbox"/>		
Use Embedded DLL	<input type="checkbox"/>	Embedded DLL Jitter Range	Low		
FIC_1 (AXI4)					
Use Initiator Interface	<input type="checkbox"/>	Use Target Interface	<input type="checkbox"/>		
Use Embedded DLL	<input type="checkbox"/>	Embedded DLL Jitter Range	Low		
FIC_2 (AXI4)					
Use Target Interface	<input type="checkbox"/>				
Use Embedded DLL	<input type="checkbox"/>	Embedded DLL Jitter Range	Low		
FIC_3 (APB)					
Use Initiator Interface	<input checked="" type="checkbox"/>				
Use Embedded DLL	<input type="checkbox"/>	Embedded DLL Jitter Range	Low		

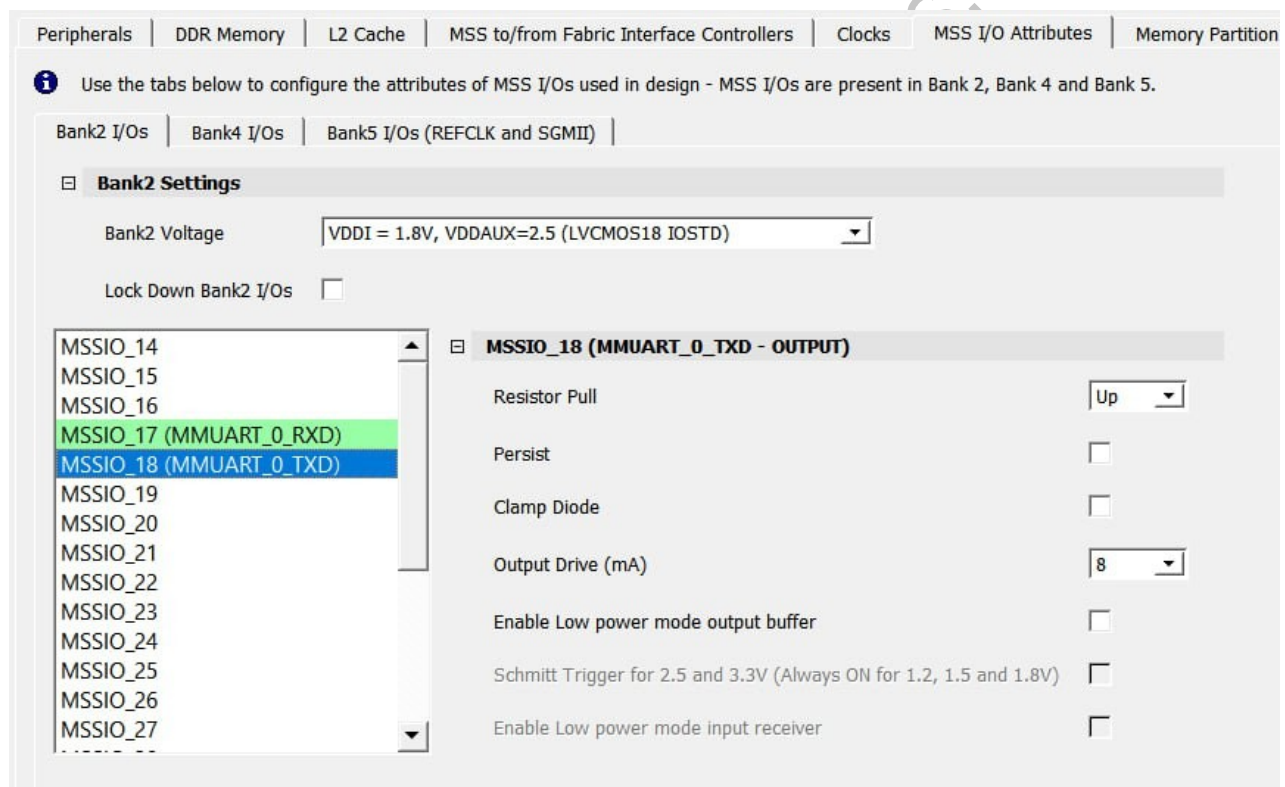
La siguiente es para la configuración de los relojes de salida del SoC, que no el de entrada que hace funcionar el SoC.

Peripherals	DDR Memory	L2 Cache	MSS to/from Fabric Interface Controllers	Clocks	MSS I/O Attributes	Memory Partition and Protection	Misc
MSS PLL and dividers							
MSS PLL reference clock source	Dedicated I/O from Bank5 (REFCLK)						
MSS PLL required clock frequency (MHz)	600						
MSS CPU cores clock frequency divider	/1						
MSS AXI clock frequency divider	/2						
MSS AHB/APB clock frequency divider	/4						
DDR							
DDR reference clock source	Dedicated I/O from Bank5 (REFCLK)						
Real Time Clock (RTC) / Gigabit Ethernet MAC							
RTC / MAC SGMII reference clock source	Dedicated I/O from Bank5 (REFCLK)						
Clock Sources Frequency							
Dedicated I/O from Bank5 (REFCLK) frequency (MHz)	125						

El SoC tiene por defecto un reloj propio, en la imagen se puede ver que tiene un reloj unido a los pines **MSS_REFCLK_IN_P/N** de 125MHz.



La siguiente es para configurar las propiedades de los pines MSS utilizados, como si están en Pull-Up, Pull-Down, etc.



La siguiente es para configurar los diferentes zonas de memoria, esto es algo complejo de manejar, debido a que el SoC cuenta con al menos 5 cores internos: 1 RV64IMAC y 4 de RV64GC.

Peripherals | DDR Memory | L2 Cache | MSS to/from Fabric Interface Controllers | Clocks | MSS I/O Attributes | Memory Partition and Protection | Misc

☐ Use Processor PMP and AXI Switch MPU Configurations

DDR Memory Partition | Processor PMP | AXI Switch MPU

☐ Setup Physical DDR Offset Address manually

	MSS Offset Address:	Range	MSS High Address:	Physical DDR Offset Address	Physical DDR High Address
Cached 1GB	0x8000_0000	128 MB	0x87FF_FFFF	0x0000_0000	0x7FF_FFFF
Cached 16GB	0x10_0000_0000	896 MB	0x10_37FF_FFFF	0x800_0000	0x3FFF_FFFF
Non-Cached 256MB	0xC000_0000	128 MB	0xC7FF_FFFF	0x4000_0000	0x47FF_FFFF
Non-Cached 16GB	0x14_0000_0000	896 MB	0x14_37FF_FFFF	0x4800_0000	0x7FFF_FFFF

Note1: Range selection for any Cached or Non-Cached memory region must be a multiple of 16 MB.
Note2: Memory is not allocated in DDR when Range is set to 0.

La última pestaña sirve para configurar unas últimas opciones del SoC, como si tiene un pin exterior para interrupciones.

Peripherals | DDR Memory | L2 Cache | MSS to/from Fabric Interface Controllers | Clocks | MSS I/O Attributes | Memory Partition and Protection | Misc

☐ **Debug Trace**

Expose MSS UltraSoC Trace ports to Fabric ☐

Expose JTAG Trace/Debug ports to Fabric ☐

Expose JTAG Trace/Debug Control via Fabric ☐

☐ **Interrupt**

Expose Interrupt ports to Fabric ☐

GPIO Interrupt register setting

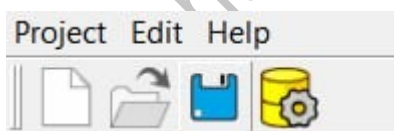
☐ **Boot Status**

Expose Boot Status ports ☐

☐ **MSS Feedback and Debug Ports**

Expose Feedback ports to Fabric ☐

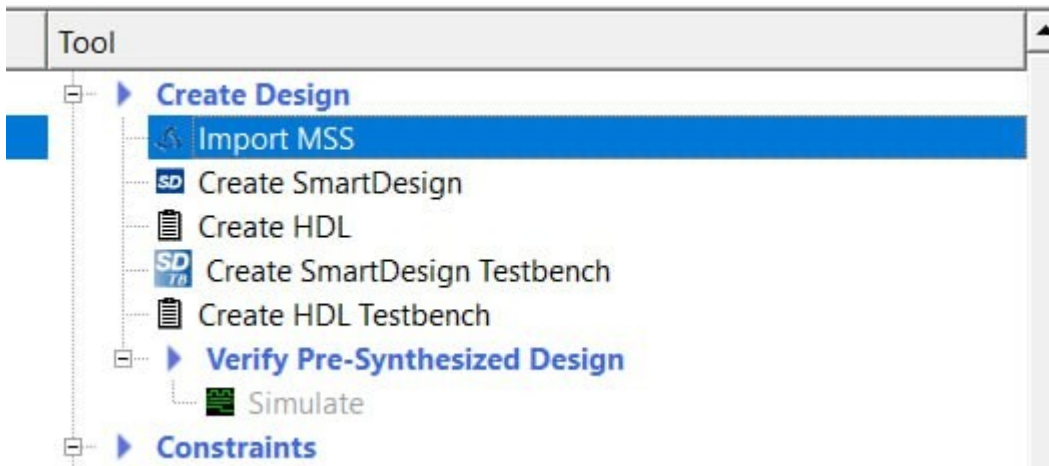
Ahora lo que se tiene que hacer es generar el componente y guardar la configuración.



Se tiene que crear un fichero .cxz para utilizarlo en Libero.

Libero

El siguiente paso es en Libero. Abrimos Libero y vamos a la opción *Import MSS*.



Esta opción nos pide un fichero .cxz como el que nos ha generado el *PFSoc MSS Configurator*.

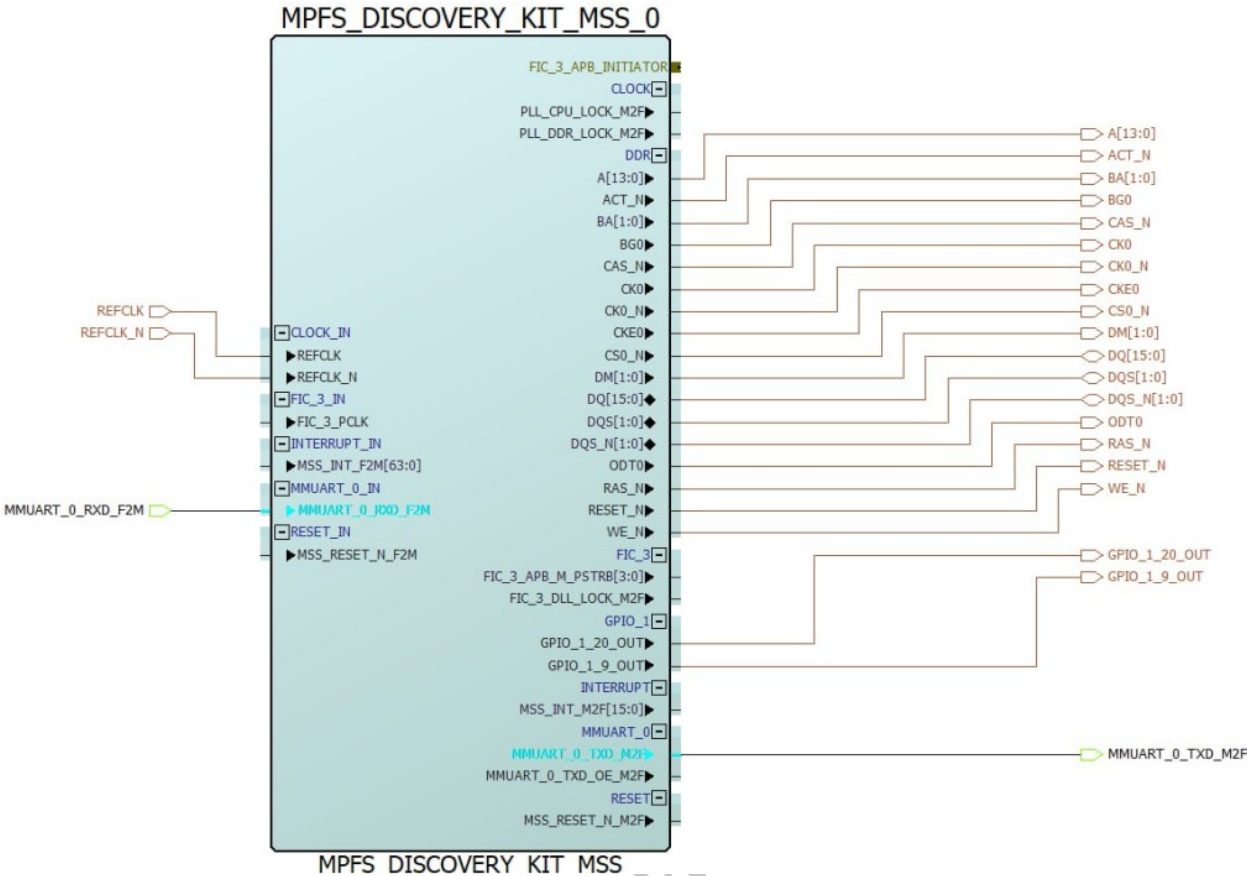
Nombre	Fecha de modificación	Tipo	Tamaño
MPFS_DISCOVERY_KIT_MSS.cxz	06/01/2025 22:43	Archivo CXZ	81 KB

Al importarlo nos aparece en la jerarquía.



Ahora solo tenemos que crear un SmartDesign y añadirlo.

Al añadirlo se puede ver que hay puertos que ya están seleccionados. Aparece un reloj diferencial, la interfaz APB seleccionada para acoplar bloques IP, también aparecen los pines de la memoria RAM, dos GPIOs seleccionados para salir por los pines MSS, y también aparece una UART con pines externos (esta UART tiene 3 pines, el pin *TXD_OE* no es necesario conectarlo).



La jerarquía queda de la siguiente manera.



Ahora lo único que falta es generar el diseño con los bloques IP deseados, o directamente generar el bitstream, y continuar con el SoftConsole.

NOTA: la UART que sale por *FABRIC*, Libero nos pide que le asignemos los pines que queremos.

GPIO_1_20_OUT	OUTPUT	LVC MOS33	B2	<input checked="" type="checkbox"/>	IOPAD_TRI
MMUART_0_RXD_F2M	INPUT	LVC MOS18	Y21	<input checked="" type="checkbox"/>	INBUF
MMUART_0_TXD_M2F	OUTPUT	LVC MOS18	W21	<input checked="" type="checkbox"/>	OUTBUF
ODT0	OUTPUT	HSTL12I	T2	<input checked="" type="checkbox"/>	IOPAD_TRI

Drivers

Los drivers de los periféricos se pueden conseguir mediante el sistema de generación interno que tiene SoftConsole 2022 (esta opción es bastante compleja de conseguir) o en este repositorio de

<https://soceame.wordpress.com/2025/01/07/configurar-los-perifericos-de-un-polarfire-soc/>

GitHub (es posible que haya drivers que no terminen de funcionar): <https://github.com/polarfire-soc/polarfire-soc-bare-metal-library>

NOTA: también se pueden conseguir algunos drivers funcionales en los ejemplos que da el SoftConsole al instalarlo, tanto para el SoftConsole 2021 como el 2022.

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