How to access IP block memory addresses in Libero

Created by: David Rubio G.

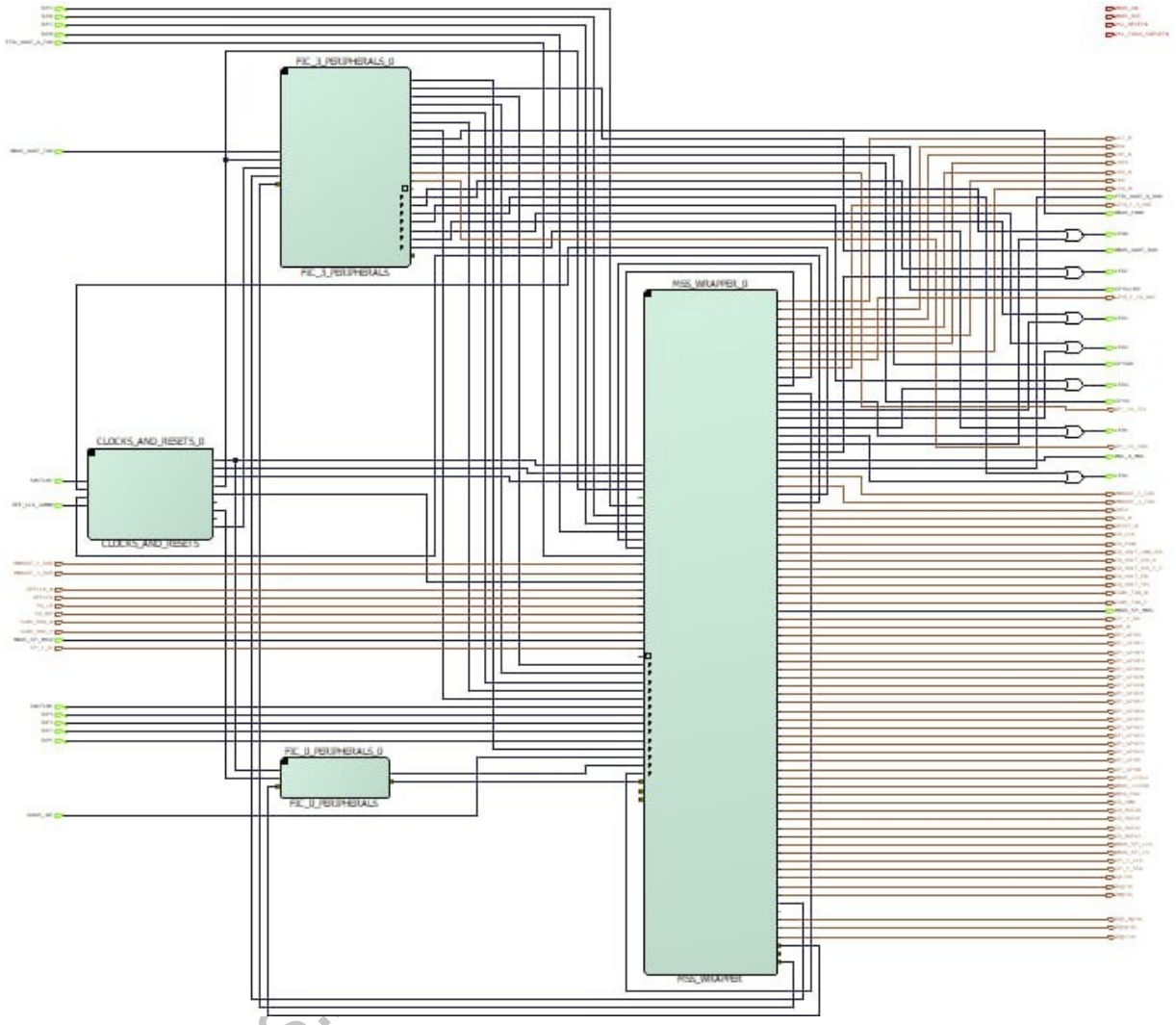
Blog post: https://soceame.wordpress.com/2025/03/11/how-to-access-ip-block-memory-addresses-in-libero/

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

GitHub: https://github.com/DRubioG

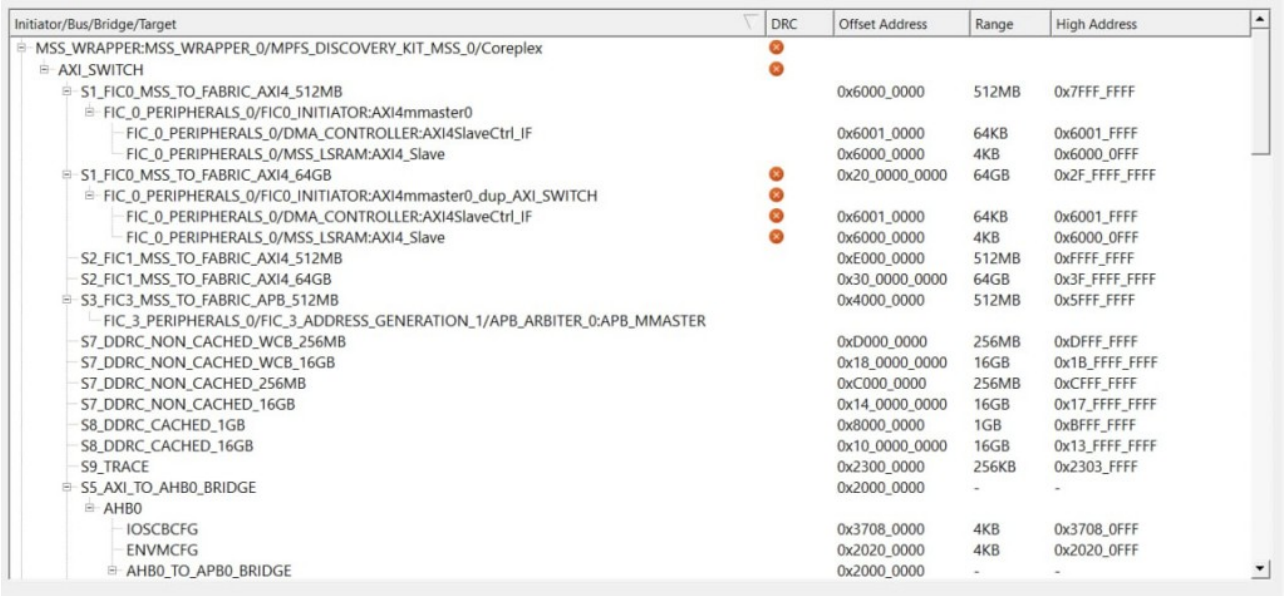
Last modification date: 11/03/25

To access the memory addresses that Libero assigns to the SmartDesign IP blocks, you must have the SmartDesign open.



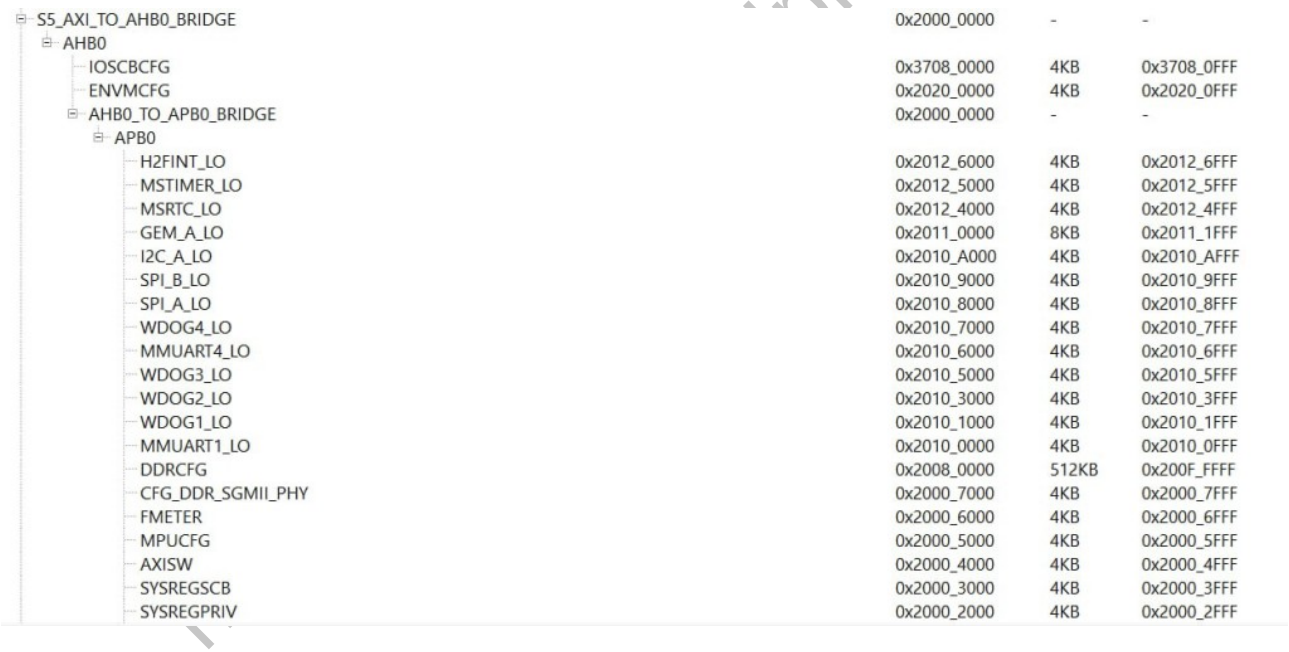
Now, all you have to do is click the following icon in the top bar.

Clicking this button opens a tab showing all the memory addresses in the desired SmartDesign. The initial memory address, the allocated memory size, and the last memory address appear.



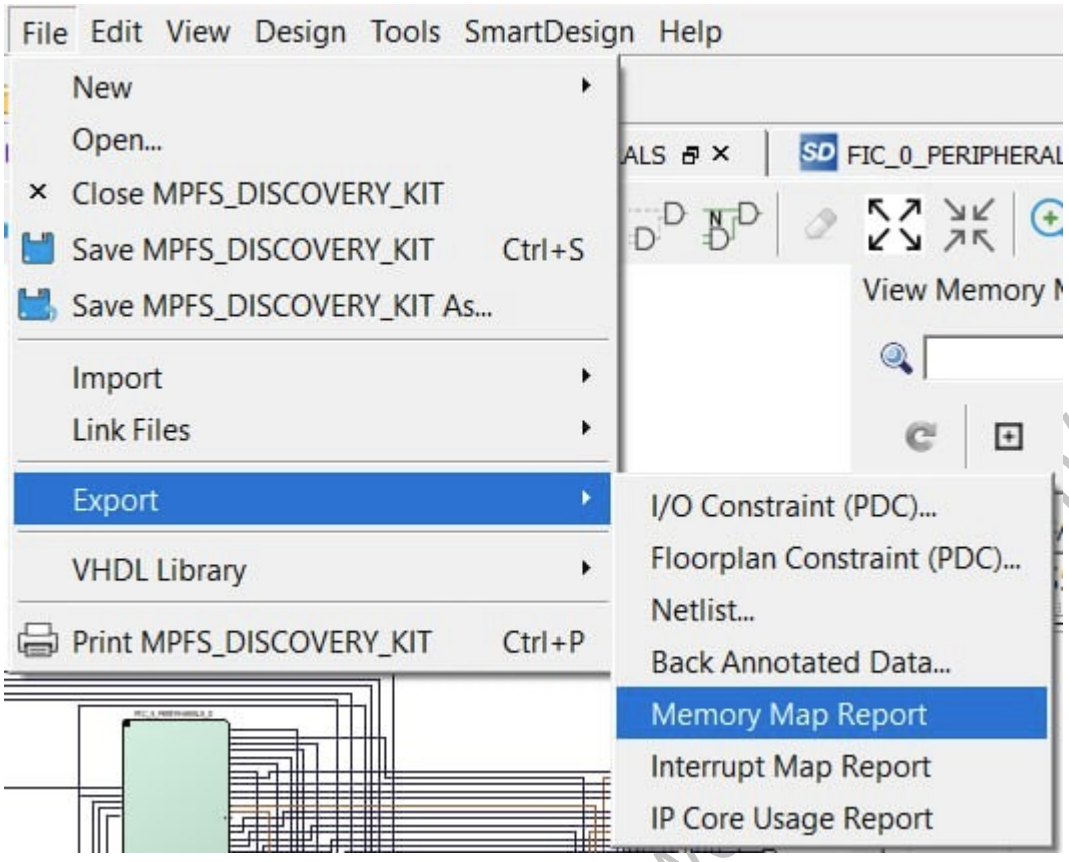
Initiator/Bus/Bridge/Target	DRC	Offset Address	Range	High Address
MSS_WRAPPER:MSS_WRAPPER_0/MPFS_DISCOVERY_KIT_MSS_0/Coreplex				
AXI_SWITCH				
S1_FIC0_MSS_TO_FABRIC_AXI4_512MB		0x6000_0000	512MB	0x7FFF_FFFF
FIC_0_PERIPHERALS_0/FIC0_INITIATOR:AXI4mmaster0				
FIC_0_PERIPHERALS_0/DMA_CONTROLLER:AXI4SlaveCtrl_IF		0x6001_0000	64KB	0x6001_FFFF
FIC_0_PERIPHERALS_0/MSS_LSRAM:AXI4_Slave		0x6000_0000	4KB	0x6000_0FFF
S1_FIC0_MSS_TO_FABRIC_AXI4_64GB		0x20_0000_0000	64GB	0x2F_FFFF_FFFF
FIC_0_PERIPHERALS_0/FIC0_INITIATOR:AXI4mmaster0_dup_AXI_SWITCH				
FIC_0_PERIPHERALS_0/DMA_CONTROLLER:AXI4SlaveCtrl_IF		0x6001_0000	64KB	0x6001_FFFF
FIC_0_PERIPHERALS_0/MSS_LSRAM:AXI4_Slave		0x6000_0000	4KB	0x6000_0FFF
S2_FIC1_MSS_TO_FABRIC_AXI4_512MB		0xE000_0000	512MB	0xFFFF_FFFF
S2_FIC1_MSS_TO_FABRIC_AXI4_64GB		0x30_0000_0000	64GB	0x3F_FFFF_FFFF
S3_FIC3_MSS_TO_FABRIC_APB_512MB		0x4000_0000	512MB	0x5FFF_FFFF
FIC_3_PERIPHERALS_0/FIC_3_ADDRESS_GENERATION_1/APB_ARBITER_0:APB_MMASTER				
S7_DDRC_NON_CACHED_WCB_256MB		0xD000_0000	256MB	0xDFFF_FFFF
S7_DDRC_NON_CACHED_WCB_16GB		0x18_0000_0000	16GB	0x1B_FFFF_FFFF
S7_DDRC_NON_CACHED_256MB		0xC000_0000	256MB	0xCFFF_FFFF
S7_DDRC_NON_CACHED_16GB		0x14_0000_0000	16GB	0x17_FFFF_FFFF
S8_DDRC_CACHED_1GB		0x8000_0000	1GB	0xBFFF_FFFF
S8_DDRC_CACHED_16GB		0x10_0000_0000	16GB	0x13_FFFF_FFFF
S9_TRACE		0x2300_0000	256KB	0x2303_FFFF
S5_AXI_TO_AHB0_BRIDGE		0x2000_0000	-	-
AHB0				
IOSCBCFG		0x3708_0000	4KB	0x3708_0FFF
ENVMCFG		0x2020_0000	4KB	0x2020_0FFF
AHB0_TO_APB0_BRIDGE		0x2000_0000	-	-

For example, it also has specific sections for peripherals connected to an interconnection block, such as an *AHB_TO_APB* to connect a data address bus (*AHB*) to a peripheral bus (*APB*).

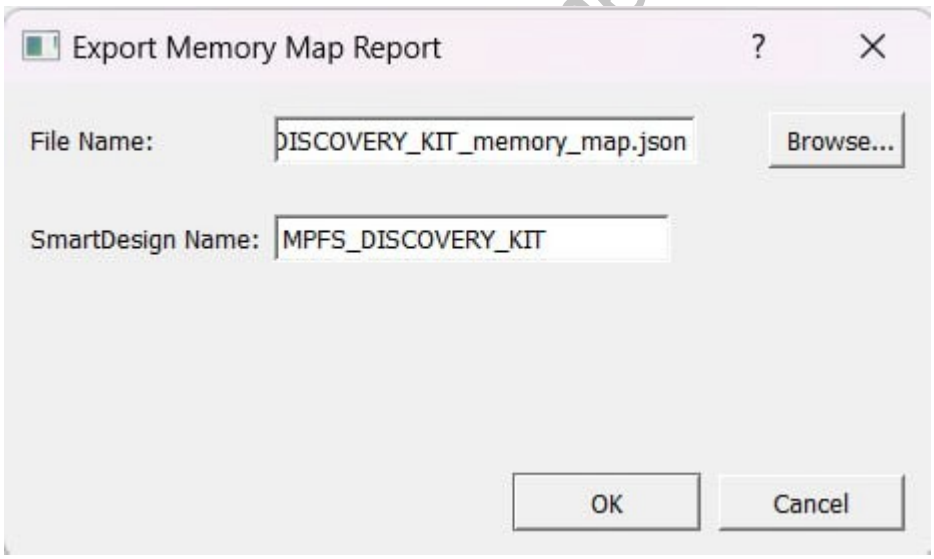


S5_AXI_TO_AHB0_BRIDGE	0x2000_0000	-	-
AHB0			
IOSCBCFG	0x3708_0000	4KB	0x3708_0FFF
ENVMCFG	0x2020_0000	4KB	0x2020_0FFF
AHB0_TO_APB0_BRIDGE	0x2000_0000	-	-
APB0			
H2FINT_LO	0x2012_6000	4KB	0x2012_6FFF
MSTIMER_LO	0x2012_5000	4KB	0x2012_5FFF
MSRTC_LO	0x2012_4000	4KB	0x2012_4FFF
GEM_A_LO	0x2011_0000	8KB	0x2011_1FFF
I2C_A_LO	0x2010_A000	4KB	0x2010_AFFF
SPI_B_LO	0x2010_9000	4KB	0x2010_9FFF
SPI_A_LO	0x2010_8000	4KB	0x2010_8FFF
WDOG4_LO	0x2010_7000	4KB	0x2010_7FFF
MMUART4_LO	0x2010_6000	4KB	0x2010_6FFF
WDOG3_LO	0x2010_5000	4KB	0x2010_5FFF
WDOG2_LO	0x2010_3000	4KB	0x2010_3FFF
WDOG1_LO	0x2010_1000	4KB	0x2010_1FFF
MMUART1_LO	0x2010_0000	4KB	0x2010_0FFF
DDRCFG	0x2008_0000	512KB	0x200F_FFFF
CFG_DDR_SGMII_PHY	0x2000_7000	4KB	0x2000_7FFF
FMETER	0x2000_6000	4KB	0x2000_6FFF
MPUCFG	0x2000_5000	4KB	0x2000_5FFF
AXISW	0x2000_4000	4KB	0x2000_4FFF
SYSREGSCB	0x2000_3000	4KB	0x2000_3FFF
SYSREGPRIV	0x2000_2000	4KB	0x2000_2FFF

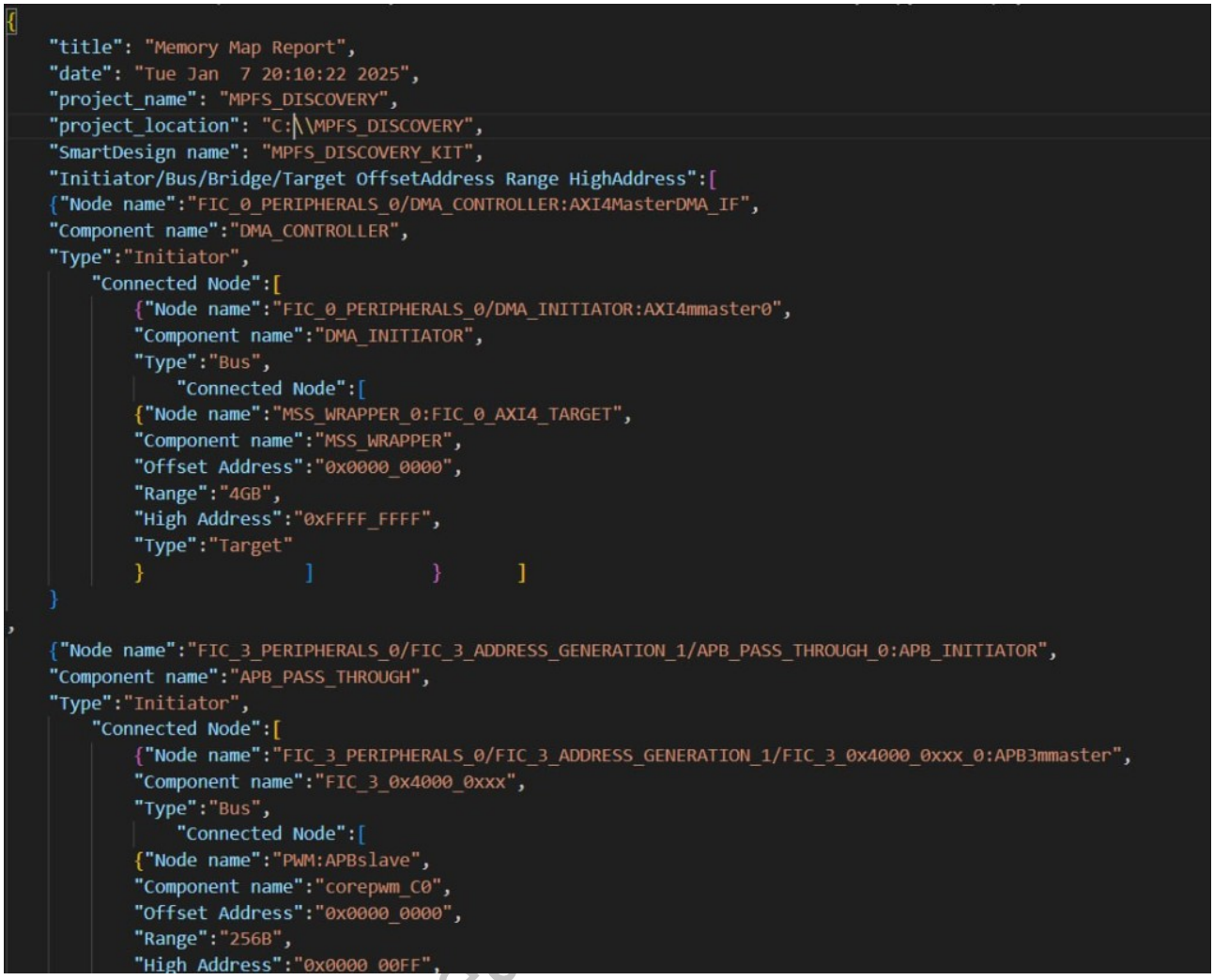
These memory addresses can also be exported.



To export it, export it as JSON.



This JSON contains all the memory address information.




```
{
    "title": "Memory Map Report",
    "date": "Tue Jan  7 20:10:22 2025",
    "project_name": "MPFS_DISCOVERY",
    "project_location": "C:\\MPFS_DISCOVERY",
    "SmartDesign name": "MPFS_DISCOVERY_KIT",
    "Initiator/Bus/Bridge/Target OffsetAddress Range HighAddress":[
    {"Node name":"FIC_0_PERIPHERALS_0/DMA_CONTROLLER:AXI4MasterDMA_IF",
    "Component name":"DMA_CONTROLLER",
    "Type":"Initiator",
        "Connected Node":[
            {"Node name":"FIC_0_PERIPHERALS_0/DMA_INITIATOR:AXI4mmaster0",
            "Component name":"DMA_INITIATOR",
            "Type":"Bus",
                "Connected Node":[
            {"Node name":"MSS_WRAPPER_0:FIC_0_AXI4_TARGET",
            "Component name":"MSS_WRAPPER",
            "Offset Address":"0x0000_0000",
            "Range":"4GB",
            "High Address":"0xFFFF_FFFF",
            "Type":"Target"
            }               ]          }       ]
    }
,
    {"Node name":"FIC_3_PERIPHERALS_0/FIC_3_ADDRESS_GENERATION_1/APB_PASS_THROUGH_0:APB_INITIATOR",
    "Component name":"APB_PASS_THROUGH",
    "Type":"Initiator",
        "Connected Node":[
            {"Node name":"FIC_3_PERIPHERALS_0/FIC_3_ADDRESS_GENERATION_1/FIC_3_0x4000_0xxx_0:APB3mmaster",
            "Component name":"FIC_3_0x4000_0xxx",
            "Type":"Bus",
                "Connected Node":[
            {"Node name":"PWM:APBslave",
            "Component name":"corepwm_C0",
            "Offset Address":"0x0000_0000",
            "Range":"256B",
            "High Address":"0x0000_00FF",
```


You can see the memory addresses used.

```
{"Node name":"I2C_A_LO",
"Offset Address":"0x2010_A000",
"Range":"4KB",
"High Address":"0x2010_AFFF",
"Type":"Target"
},
{"Node name":"SPI_B_LO",
"Offset Address":"0x2010_9000",
"Range":"4KB",
"High Address":"0x2010_9FFF",
"Type":"Target"
},
{"Node name":"SPI_A_LO",
"Offset Address":"0x2010_8000",
"Range":"4KB",
"High Address":"0x2010_8FFF",
"Type":"Target"
},
{"Node name":"WDOG4_LO",
"Offset Address":"0x2010_7000",
"Range":"4KB",
"High Address":"0x2010_7FFF",
"Type":"Target"
},
{"Node name":"MMUART4_LO",
"Offset Address":"0x2010_6000",
"Range":"4KB",
"High Address":"0x2010_6FFF",
"Type":"Target"
},
```