目录

[Contact 4](#_Toc172040794)

[SWT 4](#_Toc172040795)

[R43 5](#_Toc172040796)

[Secure roadmap 5](#_Toc172040797)

[Test command 6](#_Toc172040798)

[signal 7](#_Toc172040799)

[Doc 7](#_Toc172040800)

[AS 8](#_Toc172040801)

[Security doc 8](#_Toc172040802)

[Makefile 9](#_Toc172040803)

[Log 9](#_Toc172040804)

[主makefile 10](#_Toc172040805)

[Security 10](#_Toc172040806)

[CSSI分类 10](#_Toc172040807)

[S110 11](#_Toc172040808)

[Block diagram 11](#_Toc172040809)

[Doc 13](#_Toc172040810)

[Terminology 14](#_Toc172040811)

[Lifecycle 15](#_Toc172040812)

[MU 15](#_Toc172040813)

[S500 15](#_Toc172040814)

[Test Command 15](#_Toc172040815)

[log 16](#_Toc172040816)

[Compile 16](#_Toc172040817)

[Boot 17](#_Toc172040818)

[Trust management for NPU 18](#_Toc172040819)

[Code 18](#_Toc172040820)

[S50 19](#_Toc172040821)

[Register 19](#_Toc172040822)

[S50 cipher使用示例 24](#_Toc172040823)

[Case Porting 31](#_Toc172040824)

[EWM 31](#_Toc172040825)

[PIN\_MUX 32](#_Toc172040826)

[VIP 32](#_Toc172040827)

[MUX PTA3\_EWM0\_IN 34](#_Toc172040828)

[tb\_padi\_padring 36](#_Toc172040829)

[Tb\_padi\_mux3 46](#_Toc172040830)

[VIP 51](#_Toc172040831)

[Tb\_pad\_if 57](#_Toc172040832)

[Tb\_padi\_if 58](#_Toc172040833)

[Register 60](#_Toc172040834)

[CTRL 60](#_Toc172040835)

[COMPARE LOW 61](#_Toc172040836)

[COMPARE HIGH 61](#_Toc172040837)

[CLKPRESCALER 62](#_Toc172040838)

[Function 63](#_Toc172040839)

[Ewm\_out\_b 63](#_Toc172040840)

[SWT 64](#_Toc172040841)

[Contact 64](#_Toc172040842)

[RTL 65](#_Toc172040843)

[RESET VALUE 65](#_Toc172040844)

[SWT\_CR 65](#_Toc172040845)

[S110 66](#_Toc172040846)

[MU 66](#_Toc172040847)

[宏定义 66](#_Toc172040848)

# Contact

SE Secure architecture: Nipun Mahajan <[nipun.mahajan@nxp.com](mailto:nipun.mahajan@nxp.com)>

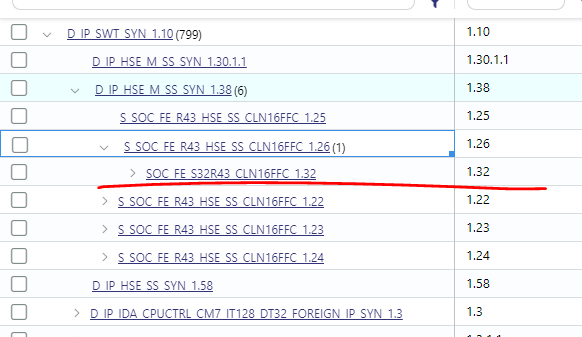
**Change to : Miroslav Knezevic**

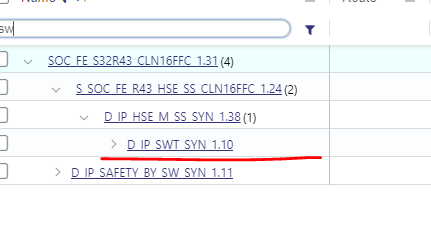
IP owner： Filippo Cioni [filippo.cioni@nxp.com](mailto:filippo.cioni@nxp.com)

S500 soc verification owner:Karan Bhardwaj [karan.bhardwaj@nxp.com](mailto:karan.bhardwaj@nxp.com)

# SWT

IP：D\_IP\_SWT\_SYN\_1.10





## R43

Cases can be refer to R43, below are the path:

/sync/publish/\_ip/fc/fc9ca7668910dd7c030c2dd4d10d0302/v\_ss\_platstim\_vip/vplan/ SWT0.csv

SWT function:

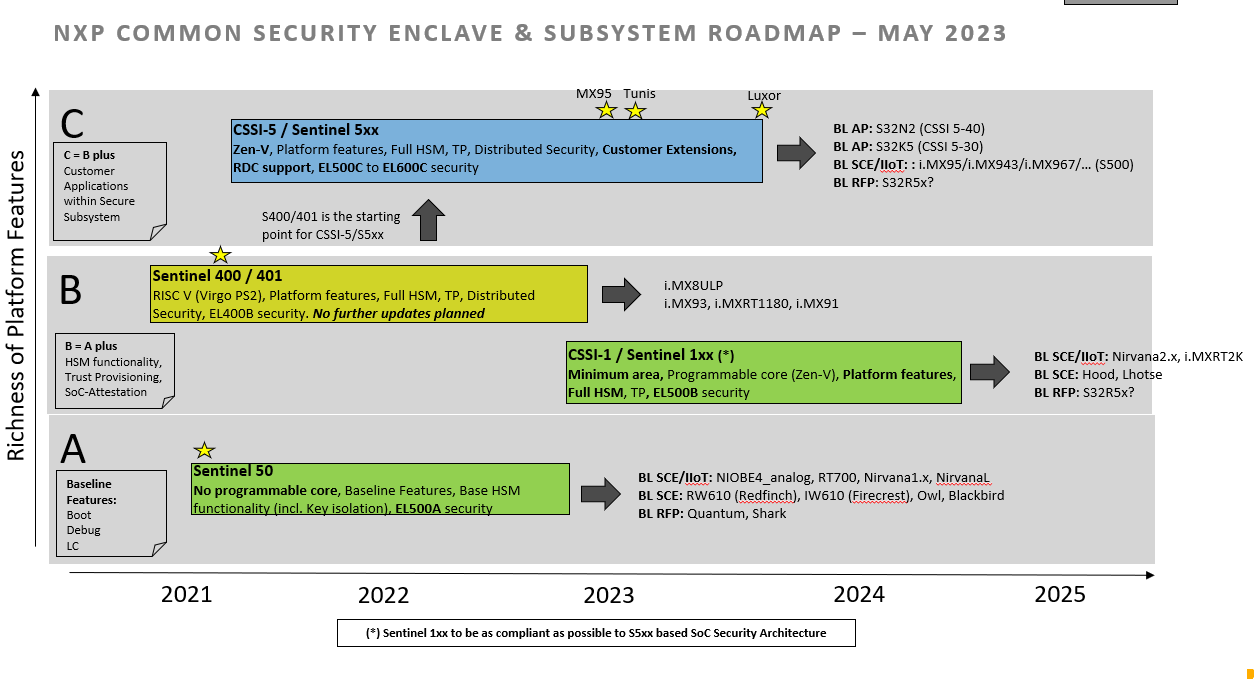
/sync/publish/\_ip/fc/fc9ca7668910dd7c030c2dd4d10d0302/v\_ss\_platstim\_vip/tool\_data/compiler/include

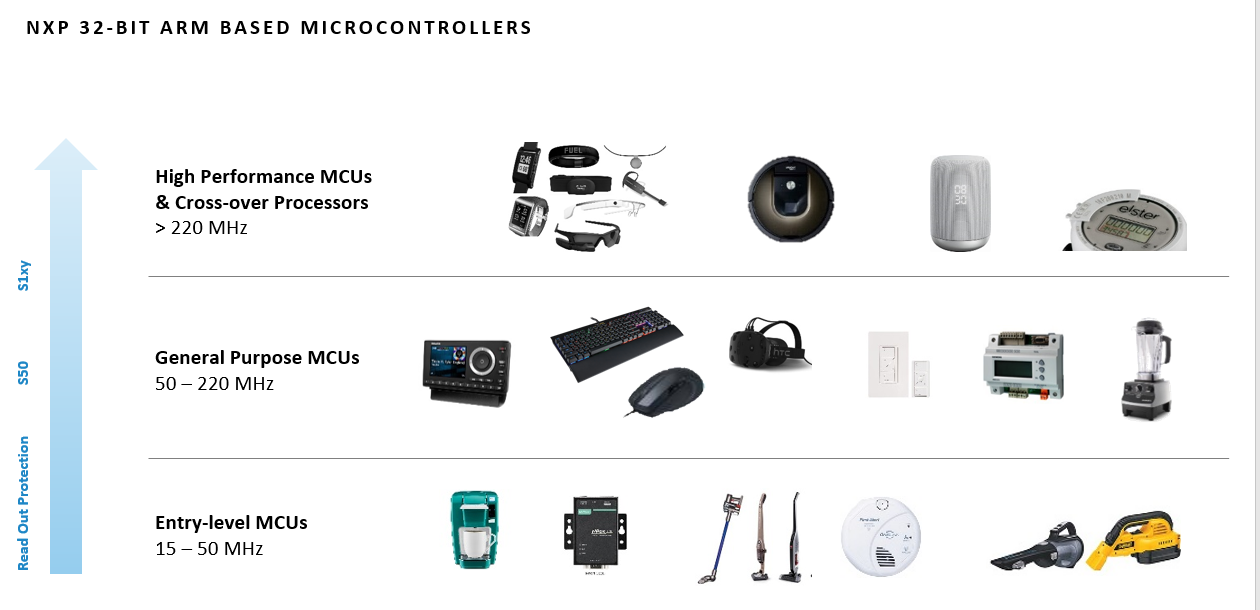
SWT stimulus

/sync/publish/\_ip/fc/fc9ca7668910dd7c030c2dd4d10d0302/v\_ss\_platstim\_vip/vectors/swt\_unified/stimulus

/home/r43\_verif/users/michael/testbench/blocks/soc\_tb/tool\_data/compiler/include

# Secure roadmap





# Test commands

For cssi sanity

bsub -R rhel7 -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors sanity -test\_name simple\_\_cssi -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -keeptemps -shm -session default&

sanity with CM85

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors sanity -test\_name simple -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -keeptemps -shm -session default&

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors sanity -test\_name simple -64bit -sim\_arg " +DELTACYCLE " -shm -session default&

开波形

进入shm文件夹，打开后缀-30以内的所有波形

bsub simvision -64bit \*-[0-9].trn \*-[0-3][0-9].trn &

EWM:

bsub -q batchq -n 3 soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors ewm -test\_name ewm\_assin\_\_inst0\_\_pin\_opt0 -64bit -sim\_arg " +DUMP\_START=800000 +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -comp\_arg " +access+rwc +sv -linedebug " -shm -session ewm-assign &

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors ewm -test\_name ewm\_reg\_\_inst0 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -comp\_arg " +access+rwc +sv -linedebug " -shm -session default&

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors ewm -test\_name ewm\_int\_\_inst0 -64bit -proj\_lib -keeptemps -comp\_arg " +access+rwc +sv -linedebug " -shm -session ewm-int&

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors ewm -test\_name ewm\_out\_\_inst0\_\_pin\_opt0 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -comp\_arg " +access+rwc +sv -linedebug " -shm -session ewm-out&

SWT:

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors swt -test\_name swt\_reg\_\_inst0 -64bit -sim\_arg " +DUMP\_START=1700000 +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -comp\_arg " +access+rwc +sv -linedebug " -shm -session swt&

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors swt -test\_name swt\_interrupt\_\_inst0 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -comp\_arg " +access+rwc +sv -linedebug " -shm -session swt-int&

CSSI

bjobs

batchq:

bsub -q batchq -n 3 soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors s110 -test\_name cssi\_reg\_mu0\_\_cm85\_\_cssi\_reg\_mu\_\_inst0\_\_cssi\_\_cssi\_reg\_mu\_\_inst0 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -shm -session cssi-porting-mu &

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors s110\_mu -test\_name cssi\_reg\_mu0\_\_cm85\_\_cssi\_reg\_mu\_\_inst0\_\_cssi\_\_cssi\_reg\_mu\_\_inst0 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -shm -session cssi-porting-mu &

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors s110 -test\_name s110\_pkc\_retention\_eien\_\_cm85\_\_s110\_pkc\_retention\_\_pdown\_\_inst0\_\_cssi\_\_s110\_pkc\_retention\_\_pdown\_\_inst0 -64bit -sim\_arg " +DUMP\_START=800000 +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -shm -session cssi-pkc\_retention-2

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors s110 -test\_name s110\_pkc\_retention\_\_cm85\_\_s110\_pkc\_retention\_\_pdown\_\_inst0\_\_cssi\_\_s110\_pkc\_retention\_\_pdown\_\_inst0 -64bit -sim\_arg " +DUMP\_START=800000 +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -shm -session cssi-pkc\_retention

## performance:

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors s110\_mu -test\_name cssi\_reg\_mu0\_\_cm85\_\_cssi\_reg\_mu\_\_inst0\_\_cssi\_\_cssi\_reg\_mu\_\_inst0 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -shm -session cssi-porting-mu &

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc xspi -vectors s110 -test\_name s110\_access\_xspi\_flash\_sha\_gcm\_32Byte\_\_cm85\_\_s110\_access\_xspi\_flash\_sha\_gcm\_32Byte\_\_inst0\_\_pin\_opt0\_\_cssi\_\_s110\_access\_xspi\_flash\_sha\_gcm\_32Byte\_\_inst0\_\_pin\_opt0 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -shm -session cssi-performance-xspi &

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors s110\_mu -test\_name cssi\_attr\_secure\_priviledge\_access\_mu0\_\_cm85\_\_cssi\_attr\_mu\_\_inst0\_\_cssi\_\_cssi\_attr\_mu.cssi\_\_inst0 -64bit -keeptemps -shm -session cssi-porting &

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors S500\_MISC -test\_name s400\_all\_ips\_\_cm33\_\_s400\_all\_ips\_\_inst0\_\_core\_s3\_\_s400\_all\_ips.s400\_\_inst0 -64bit -keeptemps -shm -session cssi-porting &

bsub -q interq -Ip soc verilog -irun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors sanity -test\_name simple\_\_cssi -64bit -keeptemps -shm -session default &

## UPF

bsub -q interq -I soc vcsmx -block soc\_tb -bc vcs\_upf -collect\_arg -tc default -vectors s110 -test\_name s110\_pkc\_retention\_eien\_\_cm85\_\_s110\_pkc\_retention\_\_pdown\_\_inst0\_\_cssi\_\_s110\_pkc\_retention\_\_pdown\_\_inst0 -proj\_lib -fsdb -session -pkc\_retention\_upf &

## ROM command

bsub -o log\_reset.txt -q xlongq soc verilog -xrun -block soc\_tb -bc rtl -collect\_arg -tc xspi -64bit -proj\_lib -no\_compile -shm -session default -comp\_arg "+access+rwc +sv +define+ZV2355\_TARMAC\_ENABLE" -sim\_arg "+trace\_inst=2 +NO\_LOOP\_TCM\_INITIAL\_DONE +ENABLE\_LFRO32K\_SPEEDUP" -make\_arg "CM85\_INST\_MEM=boot\_vector CM85\_DATA\_MEM=sram0 CM85\_STACK\_MEM=sram0 " -keeptemps -vectors pkc\_wf -test\_name RTL\_RT2660\_A0\_S40xReset-FAB-0

CSSI\_1.9 super block

bsub -q interq -I soc verilog -xrun -block soc\_tb -bc superblock -collect\_arg -tc cssi -vectors sanity -test\_name simple\_\_cssi -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -comp\_arg "+access+rwc +sv -linedebug -pkgsearchdefault -enable\_work\_yvlib" -shm -configflow -session cssi &

test\_mu\_\_cm85\_\_test\_cssi\_mu\_\_inst0\_\_cssi\_\_test\_cssi\_mu.cssi\_\_inst0.arg

bsub -q interq -I soc verilog -xrun -block soc\_tb -bc superblock -collect\_arg -tc cssi -vectors s110\_mu -test\_name cssi\_reg\_mu0\_\_cm85\_\_cssi\_reg\_mu\_\_inst0\_\_cssi\_\_cssi\_reg\_mu.cssi\_\_inst0 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -comp\_arg "+access+rwc +sv -linedebug -pkgsearchdefault -enable\_work\_yvlib" -shm -configflow -session cssi &

test：

bsub -q interq -I soc verilog -xrun -block soc\_tb -bc superblock -collect\_arg -tc cssi -vectors s110\_mu -test\_name test\_mu\_\_cm85\_\_test\_cssi\_mu\_\_inst0\_\_cssi\_\_test\_cssi\_mu.cssi\_\_inst0 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -comp\_arg "+access+rwc +sv -linedebug -pkgsearchdefault -enable\_work\_yvlib" -shm -configflow -session cssi &

SWT

bsub -q interq -I soc verilog -xrun -block soc\_tb -bc rtl -collect\_arg -tc default -vectors swt -test swt\_reg\_\_inst1 -64bit -sim\_arg " +MEMDUMP +DELTACYCLE -gui " -proj\_lib -keeptemps -comp\_arg " +access+rwc +sv -linedebug " -shm -session default &

s

# arg

修改design编译需要的IP

soc\_arg += -bc\_override "a\_ip\_pll\_usb\_480mhz\_ln28fdsoi=bhv,a\_ip\_osc\_mhz\_ln28fdsoi=uc\_svreal"

# signal

axbs:

testbench.top.main\_ss\_ft\_wrap.main\_ss.main\_axbs\_decode.sel\_slv\_m0

# Doc

RT2260 [sharepoint](https://nxp1.sharepoint.com/teams/210_34/Documents/Forms/AllItems.aspx?FolderCTID=0x0120008AAC303450669742B41A366E5B000C6E&viewid=02d95bb5%2D63ac%2D4321%2D8357%2D08f78e170813)

[Timesheet](https://nxp1.sharepoint.com/:x:/r/sites/SuzhouSoCverification/_layouts/15/Doc.aspx?sourcedoc=%7B938A6937-462E-4E97-A26A-B5D1838DB6EE%7D&file=2024_project_plan_suzhou_verification_April_9_2024.xlsx&wdLOR=c8B2E02DF-D091-40FB-8406-C1F3E3B8FFD3&action=default&mobileredirect=true)

IP Reuse：[​xlsx icon RT2660 IP Re-USE v0.1.xlsx](https://nxp1.sharepoint.com/:x:/r/teams/210_34/Documents/04_Design/IP%20Reuse%20Table/RT2660%20IP%20Re-USE%20v0.1.xlsx?d=wba626b09c72f49aebbe5ff2a96ef4bcd&csf=1&web=1&e=onXi7G),



## AS

[​pdf icon 20240212-RT2660\_HW\_AS 0v3\_review.pdf](https://nxp1.sharepoint.com/:b:/r/teams/210_34/Documents/05_Systems/Architecture%20Spec/20240212-RT2660_HW_AS%200v3_review.pdf?csf=1&web=1&e=yhj1kO)

## Security doc

training： [Security Awareness Training Advanced](https://nww.sharepoint.nxp.com/teams/97/SitePages/Security%20Awareness%20Training.aspx)

ELM: [Link](https://elm.nxp.com/rm/web#action=com.ibm.rdm.web.pages.showArtifactPage&artifactURI=https%3A%2F%2Felm.nxp.com%2Frm%2Freviews%2F_mdJrAeVcEe67J74haR_uXA&componentURI=https%3A%2F%2Felm.nxp.com%2Frm%2Frm-projects%2F_YVnvEJuvEe6SD7AeCwYG7Q%2Fcomponents%2F_YaGmcJuvEe6SD7AeCwYG7Q&vvc.configuration=https%3A%2F%2Felm.nxp.com%2Frm%2Fcm%2Fbaseline%2F_PmTm0OTzEe67J74haR_uXA) CSSI AS was in ELM system

Seclore: [Link](https://nxp.stringsend.com/nxp/) IMX943 RM was in Seclore

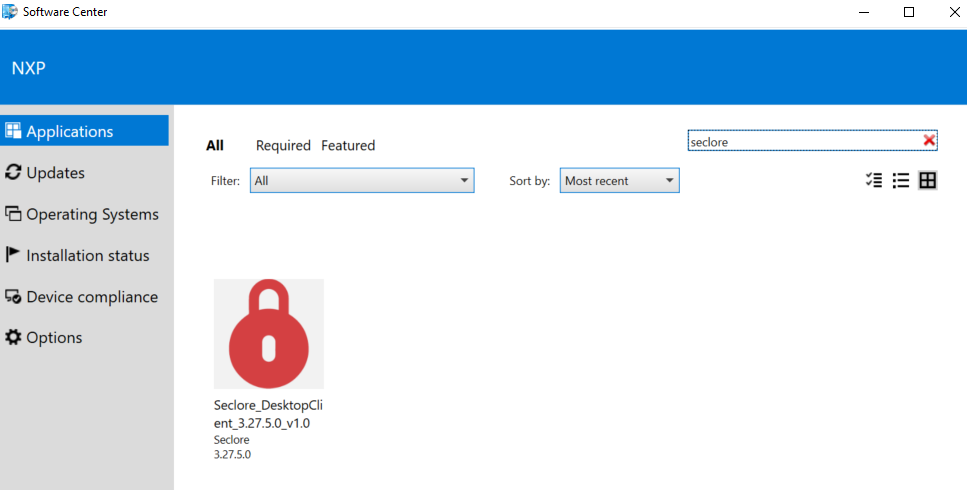
Soc IG guide： [CSSI soc support doc](https://nww.amec.sharepoint.nxp.com/teams/248_3/CSSI/Forms/AllItems.aspx?FolderCTID=0x01200025BEBFE641975E49B4D1230D3AB3F49A&id=%2Fteams%2F248%5F3%2FCSSI%2FDocs%2FSoC%20Support%20Docs%2FCSSI)

RM： [CSSI\_Seclore\_doc](https://nww.amec.sharepoint.nxp.com/teams/248_3/Seclore%20Docs/Forms/AllItems.aspx?RootFolder=%2fteams%2f248%5f3%2fSeclore%20Docs%2fCSSI&FolderCTID=0x01200001937A3F47FFF44F977204EE7B8D2C4A) （需要安装seclore）

Install Seclore app in windows:



1. Request app in software center [link](https://nxp2.service-now.com/sp?id=sc_cat_item&table=sc_cat_item&sys_id=e940565487040a98000adcaabbbb353a&recordUrl=com.glideapp.servicecatalog_cat_item_view.do%3Fv%3D1&sysparm_id=e940565487040a98000adcaabbbb353a), fill the device name
2. Once request approved, install Seclore



# Makefile

Imx95：

Log：

/home/imx95\_verif\_nobackup/usr/nxa28190/imx95\_b0/vcsmx\_rtl\_unit\_default/cssi\_boot/\_SOC/s400\_single\_boot\_flow\_\_cm33\_\_s400\_single\_boot\_flow\_\_inst0\_\_core\_s3\_\_s400\_single\_boot\_flow.s400\_\_inst0

Waveform:

/home/imx95\_verif\_nobackup/usr/nxa28190/imx95\_b0/vcsmx\_rtl\_unit\_default/cssi\_boot/s400\_single\_boot\_flow\_\_cm33\_\_s400\_single\_boot\_flow\_\_inst0\_\_core\_s3\_\_s400\_single\_boot\_flow.s400\_\_inst0VrtlCunitM.fsdb

## Log

# | /home/imx95\_verif/usr/nxa28190/imx95\_b0/testbench/blocks/soc\_tb/tool\_data/compiler/include/c\_mem\_map.h

# | /home/imx95\_verif\_nobackup/usr/nxa28190/imx95\_b0/vcsmx\_rtl\_unit\_default/default.linux26\_64bit/mem\_map.v

### Target CPU : cm33\_s

### Target CPU Type : cm33

### Target CPU : zenv

### Target CPU Type : zenv

/home/imx95\_verif/usr/nxa28190/imx95\_b0/testbench/common\_blocks/v\_ip\_makefile\_zenv/tool\_data/compiler/makefile.stimulus.compile

## 主makefile

/home/imx95\_verif/usr/nxa28190/imx95\_b0/testbench/blocks/soc\_tb/testbench/makefile.stimulus

# Security

S110 = CSSI\_1\_10

Common Security SubSystem IP (CSSI)

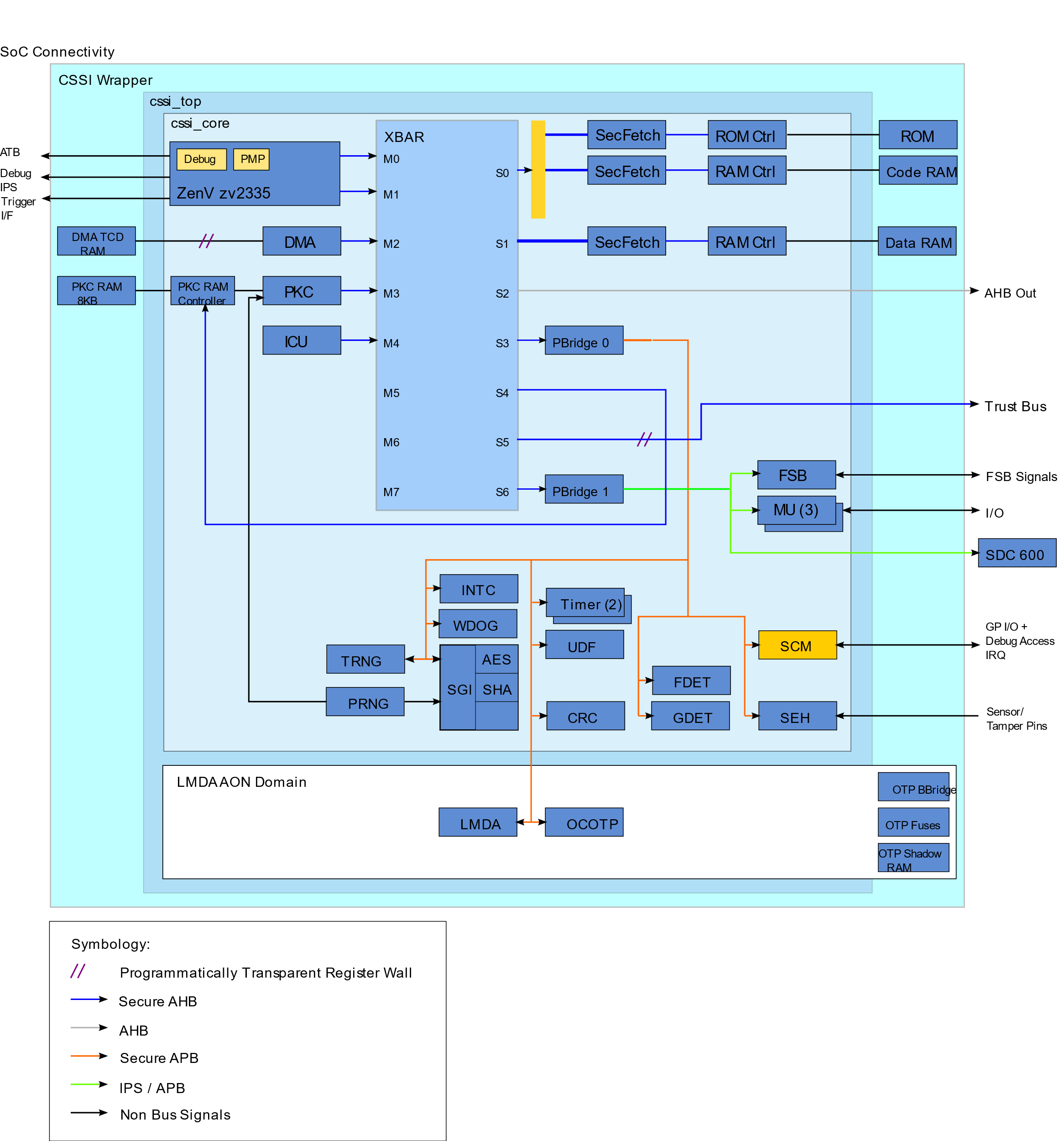
S50- CSS = Crypto Secure Sequencer

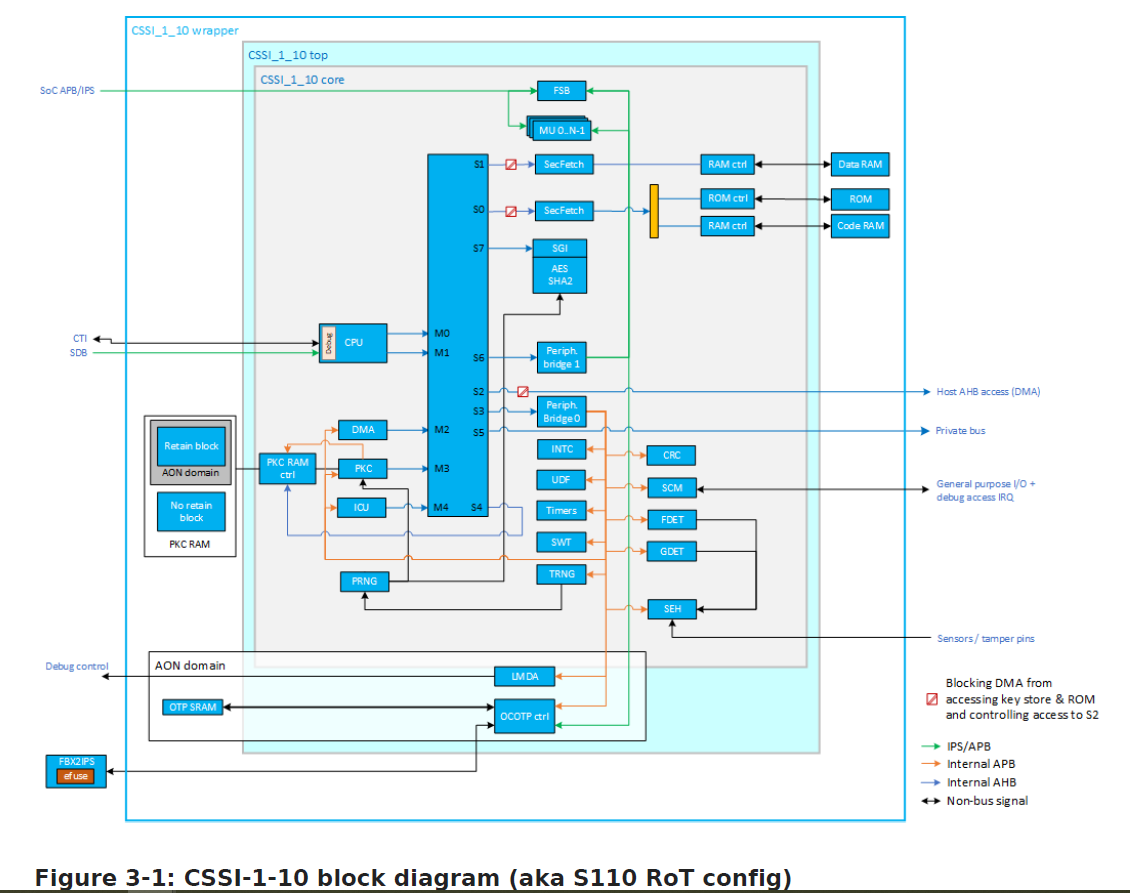
## CSSI分类

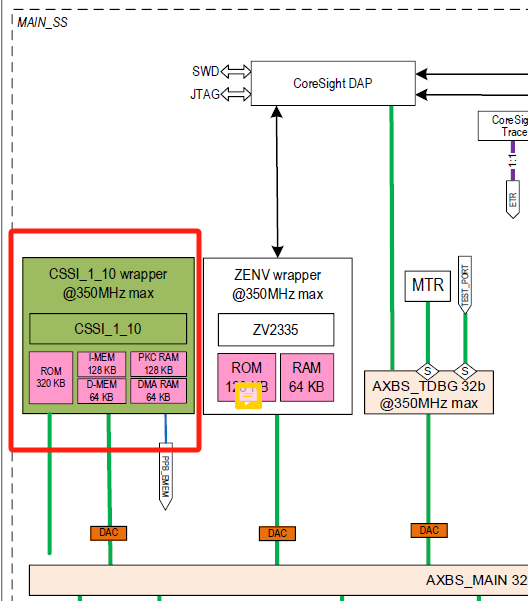


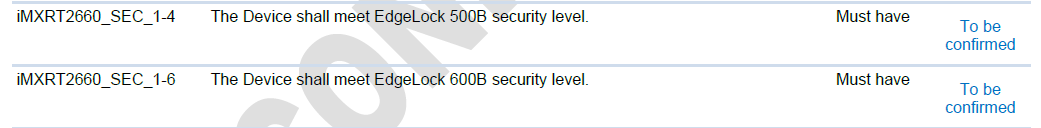
## S110

### Block diagram





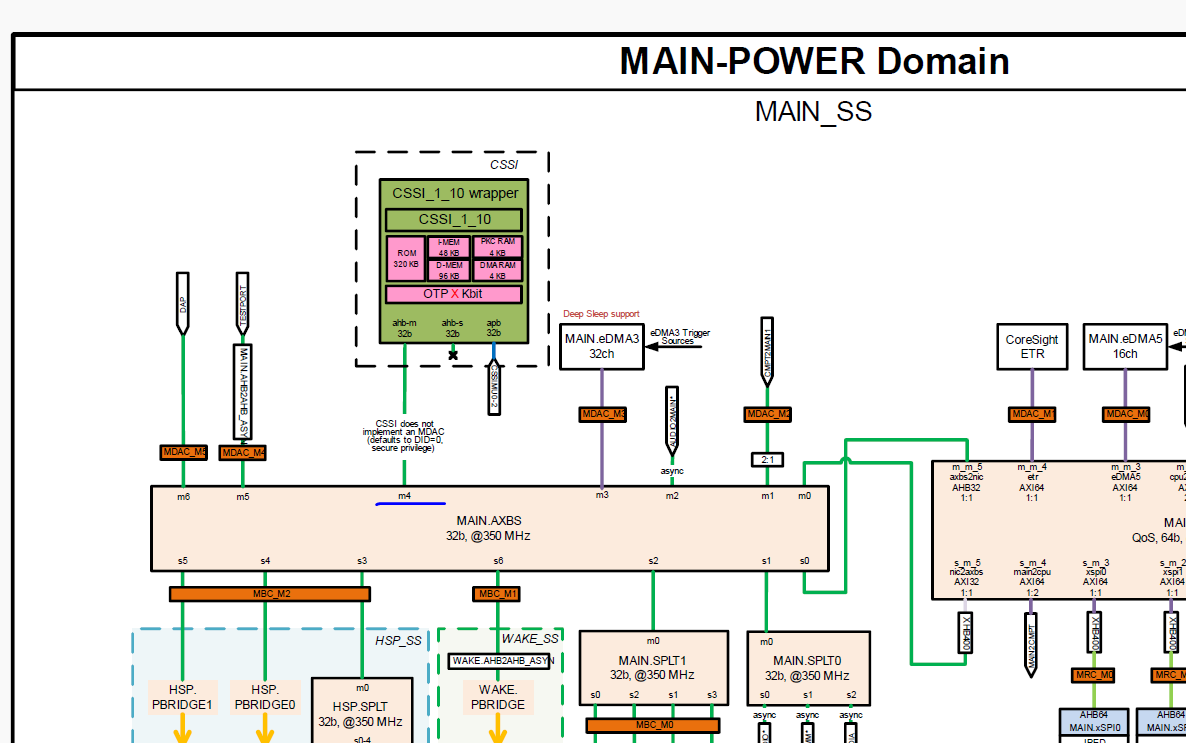




#### SOC connection

S110 was resident in main SS

S110 access SOC resources by AHB M4 port.



### Doc

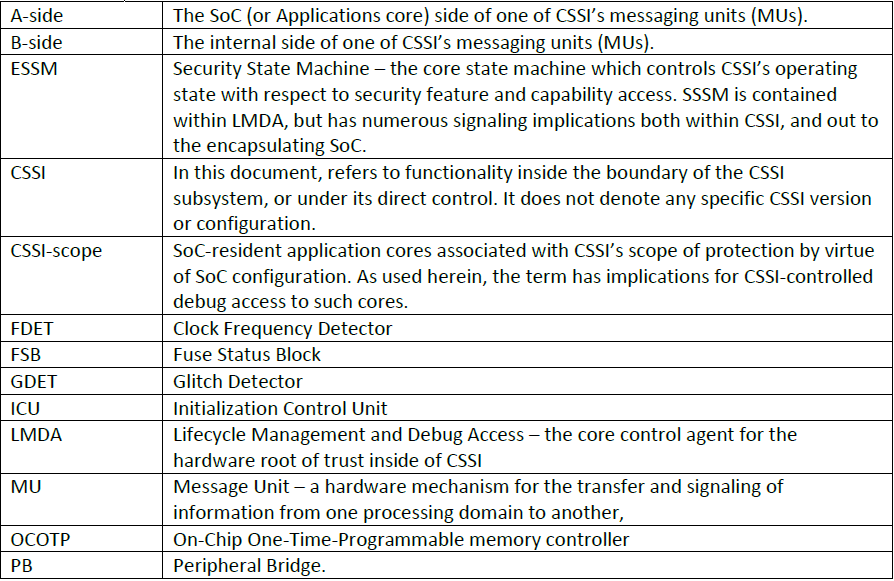
[CSSI](https://nww.amec.sharepoint.nxp.com/teams/248_3/CSSI/Forms/AllItems.aspx?RootFolder=%2fteams%2f248%5f3%2fCSSI%2fDocs%2fSoC%20Support%20Docs&FolderCTID=0x01200025BEBFE641975E49B4D1230D3AB3F49A) User guide/Integration guide

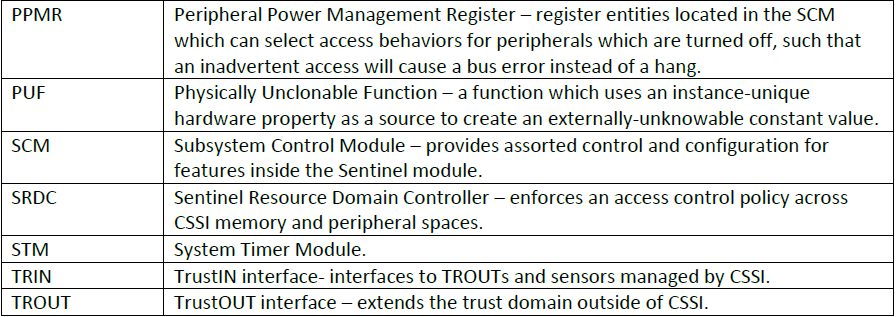
* Specs:

[Common Security Subsystem IP (CSSI) AS part 1](https://elm.nxp.com/rm/resources/MD_90ZMYJ3NEe6SD7AeCwYG7Q?vvc.configuration=https%3A%2F%2Felm.nxp.com%2Frm%2Fcm%2Fbaseline%2F_PmTm0OTzEe67J74haR_uXA)  
[Common Security Subsystem IP (CSSI) AS part 2](https://elm.nxp.com/rm/resources/MD_ClIXgJ3OEe6SD7AeCwYG7Q?vvc.configuration=https%3A%2F%2Felm.nxp.com%2Frm%2Fcm%2Fbaseline%2F_PmTm0OTzEe67J74haR_uXA)  
[Common Security Subsystem IP (CSSI) RS](https://elm.nxp.com/rm/resources/MD_vHvf0J3PEe6SD7AeCwYG7Q?vvc.configuration=https%3A%2F%2Felm.nxp.com%2Frm%2Fcm%2Fbaseline%2F_PmTm0OTzEe67J74haR_uXA)

* Access the Review: [RT2660 S110 HW AS review](https://elm.nxp.com/rm/reviews/_mdJrAeVcEe67J74haR_uXA)

### Terminology





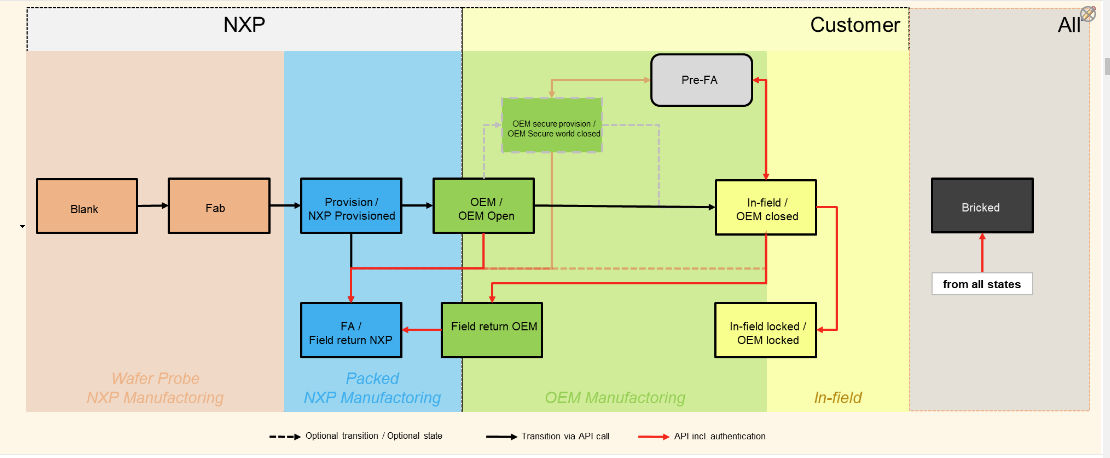
### Memory map

|  |  |  |  |
| --- | --- | --- | --- |
| Start Address | End Address | Port | Slave Description |
| 0x0000\_0000 | 0xdfff\_ffff | S2 | External output to SoC |
| 0xe000\_0000 | 0xe004\_ffff | S0 | ROM 320K-bytes |
| 0xe7fe\_0000 | 0xe7ff\_ffff | S0 | IMEM 96Keg-bytes |
| 0xe800\_0000 | 0xe800\_bfff | S1 | DMEM 48K-bytes |
| 0xeb00\_0000 | 0xeb00\_1fff | S4 | DMEM PKC (PKC) 8K-bytes |
| 0xec00\_0000 | 0xec05\_ffff | S3 | Internal Peripherals \*\* |
| 0xed00\_0000 | 0xed03\_ffff | S6 | SoC-interconnected Peripherals \*\* |
| 0xee00\_0000 | 0xefff\_ffff | S5 | TrustBus |
| 0xf000\_0000 | 0xffff\_ffff |  | Reserved |

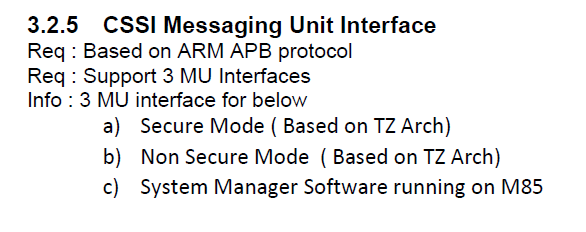
A diagram of a computer

Description automatically generated

### Lifecycle



### MU



## S500

### Test Command

进入如下目录/home/mx943\_verif/usr/richard\_nxa28190/1.45/testbench/blocks/soc\_tb/testbench

make vcsmx\_rtl\_comp IMX95=yes vcsmx\_rtl\_sim vectors=cssi\_boot tc=default testname=s400\_single\_boot\_flow\_\_cm33\_\_s400\_single\_boot\_flow\_\_inst0\_\_core\_s3\_\_s400\_single\_boot\_flow.s400\_\_inst0 args=-sim\_arg\ +dontStopOnSimulError=1

imx943:

bsub -R rhel7 -q interq -Ip make vcsmx\_rtl\_sentinel\_comp vcsmx\_rtl\_sentinel\_sim vectors=s500\_mu testname=s400\_reg\_mu0\_\_cm33\_\_s400\_reg\_mu\_\_inst0\_\_core\_s3\_\_s400\_reg\_mu.s400\_\_inst0 XPROP=0 args=-sim\_arg\ +dontStopOnSimulError=1 &

Verdi -ssf

### log

/home/imx95\_verif\_nobackup/usr/nxa28190/imx95\_b0/vcsmx\_rtl\_unit\_sentinel/cssi\_boot/vcsmx\_FILENAMES\_apcw401i-10-60-37-117.nxdi.kr-awsc1.nxp.com\_65464/makefile.tbc

INFO: Creating file

/home/imx95\_verif\_nobackup/usr/nxa28190/imx95\_b0/vcsmx\_rtl\_unit\_sentinel/cssi\_boot/vcsmx\_FILENAMES\_apcw401i-10-60-37-117.nxdi.kr-awsc1.nxp.com\_65464/runsim.sh

INFO: Creating file

/home/imx95\_verif\_nobackup/usr/nxa28190/imx95\_b0/vcsmx\_rtl\_unit\_sentinel/cssi\_boot/vcsmx\_FILENAMES\_apcw401i-10-60-37-117.nxdi.kr-awsc1.nxp.com\_65464/runall.sh

/home/imx95\_verif\_nobackup/usr/nxa28190/imx95\_b0/vcsmx\_rtl\_unit\_sentinel/cssi\_boot/\_SOC/s400\_single\_boot\_flow\_\_cm33\_\_s400\_single\_boot\_flow\_\_inst0\_\_core\_s3\_\_s400\_single\_boot\_flow.s400\_\_inst0/soc\_vcsmx\_s400\_single\_boot\_flow\_\_cm33\_\_s400\_single\_boot\_flow\_\_inst0\_\_core\_s3\_\_s400\_single\_boot\_flow.s400\_\_inst0.log

/home/mx943\_verif\_nbk/usr/richard\_nxa28190/1.45/vcsmx\_rtl\_unit\_sentinel/s500\_mu/logfiles

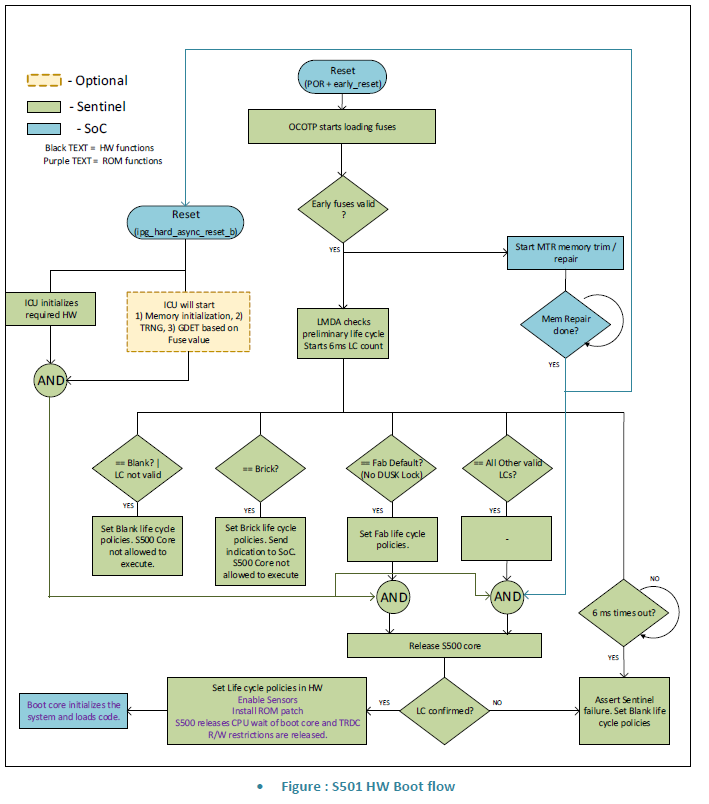
/home/mx943\_verif\_nbk/usr/richard\_nxa28190/1.45/vcsmx\_rtl\_unit\_sentinel/s500\_mu/results/s400\_reg\_mu0\_\_cm33\_\_s400\_reg\_mu\_\_inst0\_\_core\_s3\_\_s400\_reg\_mu.s400\_\_inst0VrtlCunitM.fsdb

### Compile

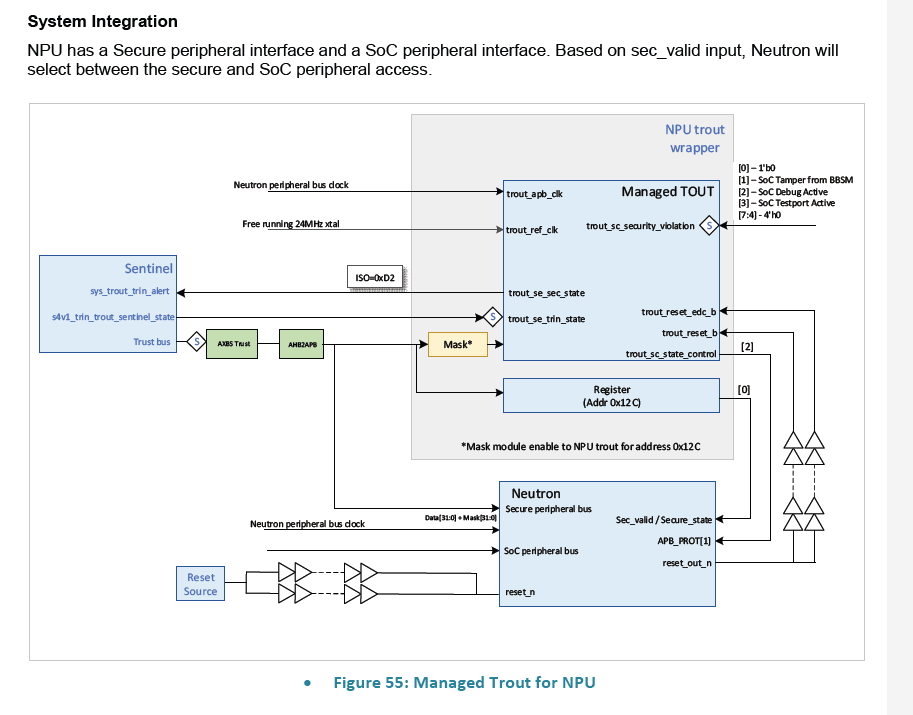
arm-none-eabi-as

  CC            = riscv32-unknown-elf-g++  
  AS            = riscv32-unknown-elf-as  
  LD            = riscv32-unknown-elf-ld  
  NM            = riscv32-unknown-elf-nm  
  OBJDUMP       = riscv32-unknown-elf-objdump  
  OBJCOPY       = riscv32-unknown-elf-objcopy  
  CPP           = riscv32-unknown-elf-cpp

### Boot



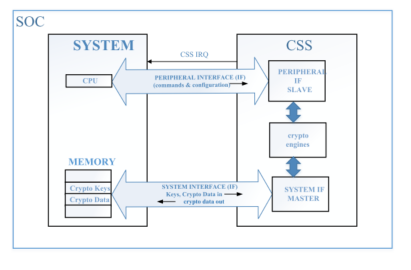
### Trust management for NPU

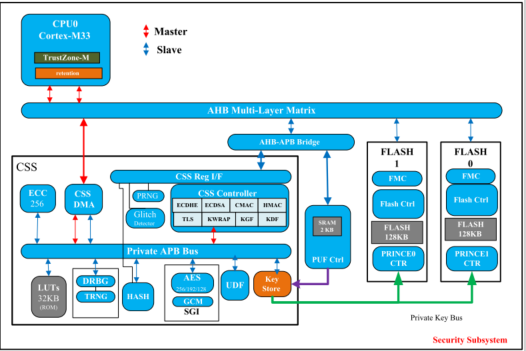


### Code

/home/imx95\_verif/usr/nxa28190/imx95\_b0/testbench/blocks/soc\_tb/tool\_data/compiler/zenv/zenv/include/pre\_post\_main.h

## S50

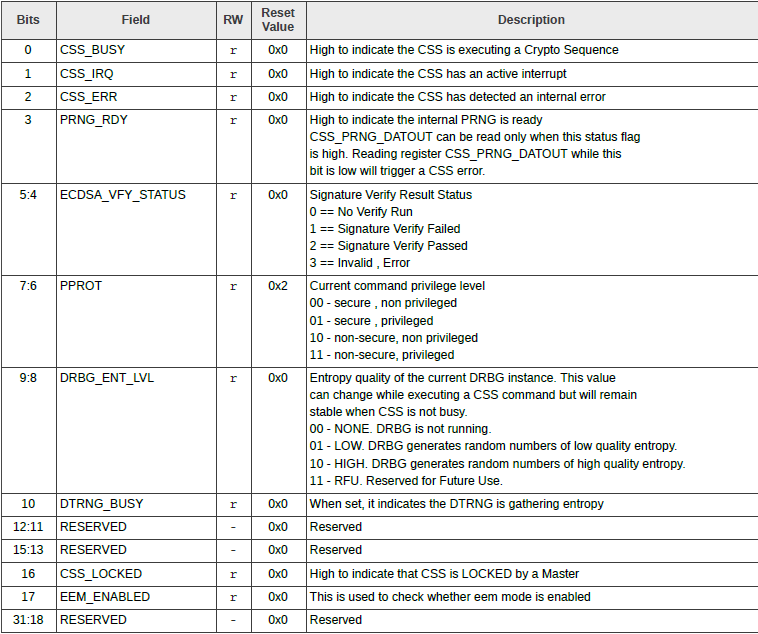




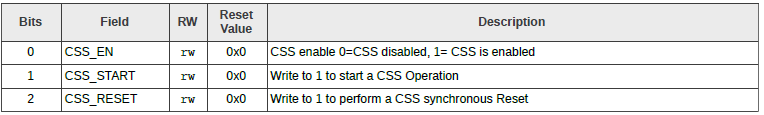
### Register

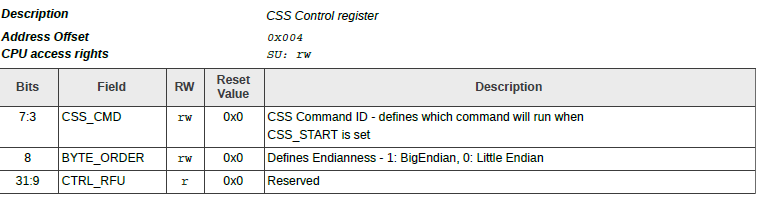
#### CSS\_STATUS

Offset 0x0：

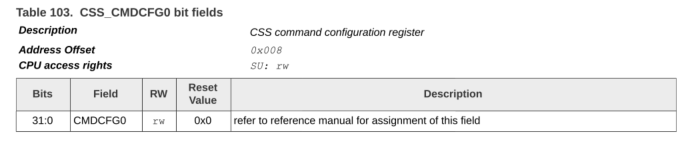


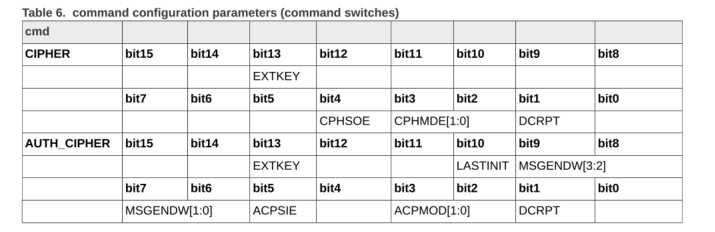
#### CSS\_CTRL（enable/start）



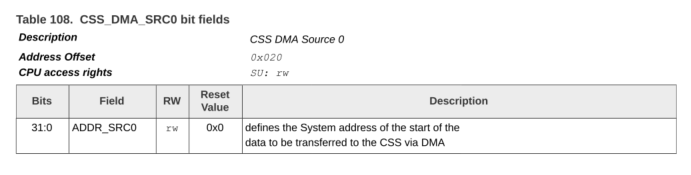


#### CSS\_CMDCFG0

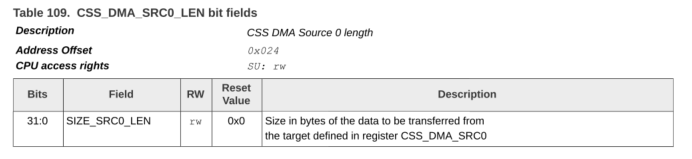




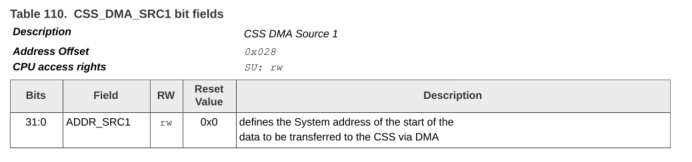
#### CSS\_DMA\_SRC0



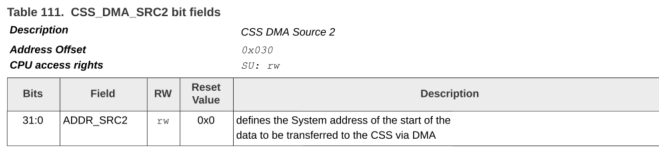
#### CSS\_DMA\_SCR0\_LEN



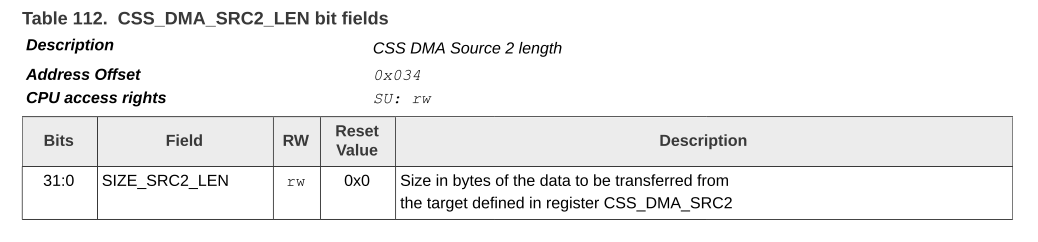
#### CSS\_DMA\_SRC1



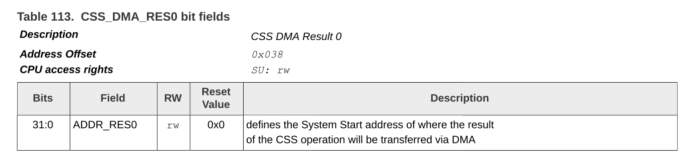
#### CSS\_DMA\_SRC2



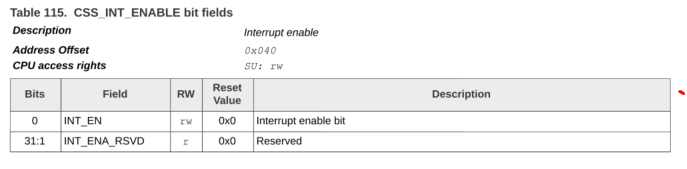
#### CSS\_DMA\_SRC2\_LEN



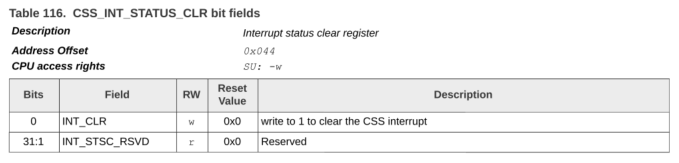
#### DMA\_CSS\_RES0



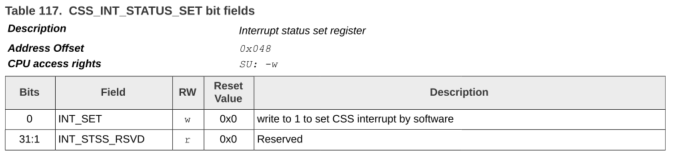
#### CSS\_INT\_ENABLE



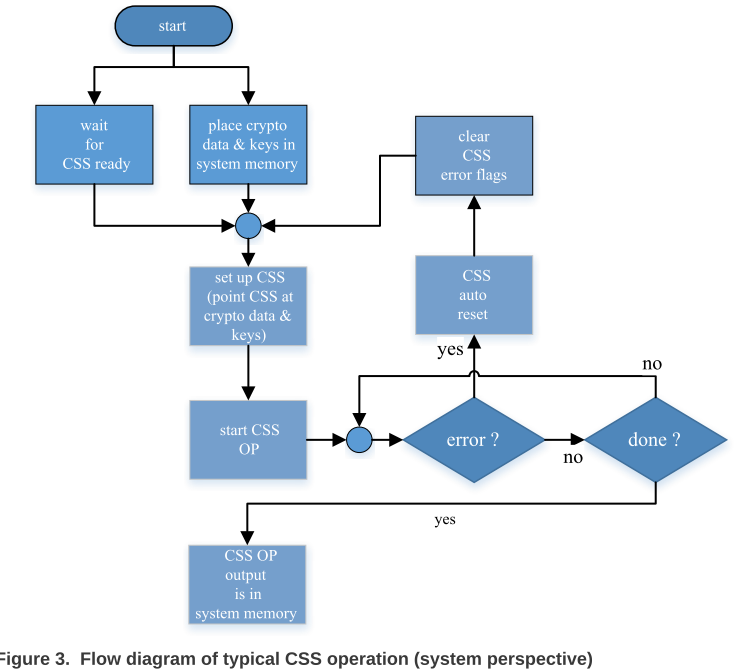
#### CSS\_INT\_STATUS\_CLR



#### CSS\_INT\_STATUS\_SET



### S50 cipher使用示例



#### Enable CSS clock

// Enable CSS clock

write\_syscon\_cmpt\_els\_clk\_ctrl\_els\_clk\_en(1);

write\_syscon\_cmpt\_els\_clk\_ctrl\_dtrng\_refclk\_en(1);

W32: A=50190004 W=00000001

#### Initialize CSS

// - Write CSS\_EN bit and wait for BUSY bit to go low

cssv2\_init(cssv2\_context);

##### enable/等待busy bit清零

[W32: A=50190004 W=00000001](#_CSS_CTRL（enable/start）)

[R32: A=50190000 R=00000482](#_CSS_STATUS)

##### 清除中断

// Clear the first interrupt //

cssv2\_clear\_int(cssv2\_context);

W32(CSS\_ERR\_STATUS\_CLR,BIT0);//clear inital dtrng error//according to the spec, it is not a css error

[W32: A=50190044 W=00000001](#_CSS_INT_STATUS_CLR)

##### 使能中断

// Enable cssv2 interrupt

cssv2\_enable\_int(cssv2\_context);

[BSET32: A=50190040 Ri=00000000 M=00000001 Rf=00000001](#_CSS_INT_ENABLE)

#### 系统memory中写入待加解密数据/密钥/IV

// Plaintext

for (mem\_idx = 0; mem\_idx < (PTX\_SIZE / 4); mem\_idx++) {

W32(REG32((PTX\_PTR + (mem\_idx << 2) + 0 + tv\_idx)) , ((plaintext[tv\_idx][mem\_idx] >> 0 ) & 0xFF));

W32(REG32((PTX\_PTR + (mem\_idx << 2) + 1 + tv\_idx)) , ((plaintext[tv\_idx][mem\_idx] >> 8 ) & 0xFF));

W32(REG32((PTX\_PTR + (mem\_idx << 2) + 2 + tv\_idx)) , ((plaintext[tv\_idx][mem\_idx] >> 16) & 0xFF));

W32(REG32((PTX\_PTR + (mem\_idx << 2) + 3 + tv\_idx)) , ((plaintext[tv\_idx][mem\_idx] >> 24) & 0xFF));

}

// IV

for (mem\_idx = 0; mem\_idx < (IV\_SIZE / 4); mem\_idx++) {

W32(REG32((IV\_PTR + (mem\_idx << 2))) , iv[tv\_idx][mem\_idx]);

}

// KEY

for (mem\_idx = 0; mem\_idx < (KEY\_SIZE / 4); mem\_idx++) {

W32(REG32((KEY\_PTR + (mem\_idx << 2))) , key[tv\_idx][mem\_idx]);

}

#### 配置数据到CSS DMA中

R32(CSS\_STATUS,rdata);

SHOW32(STIMNAME,"CSS alias window = 0x%08x", rdata + (tv\_idx << 10)); // (tv\_idx << 10) is offset count in word, not in byte

##### 待加解密数据写入SRC0

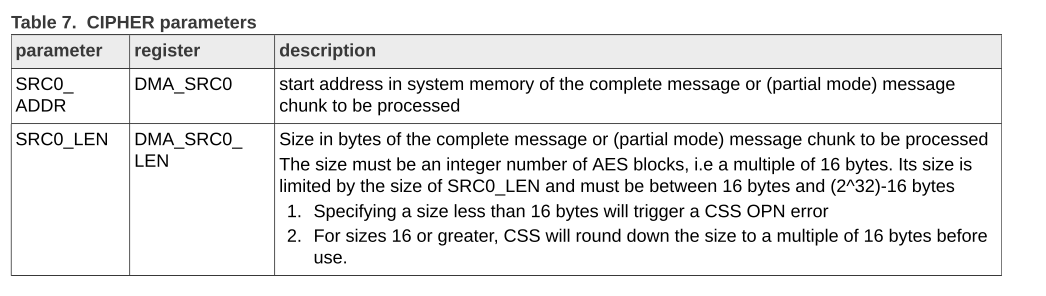
// Load pointer to Plaintext

cssv2\_load\_dma\_src0(cssv2\_context, PTX\_PTR + tv\_idx); //TODO need +tv\_idx?

cssv2\_load\_dma\_src0\_len(cssv2\_context, PTX\_SIZE);

[W32: A=50190020 W=30000800](#_CSS_DMA_SRC0)

[A=50190024 W=00000020](#_CSS_DMA_SCR0_LEN)

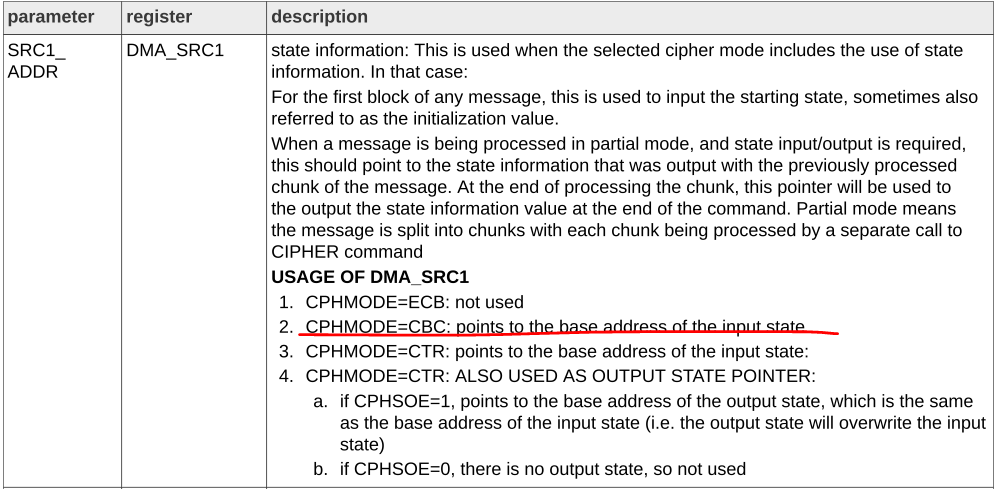


##### IV 地址写入SRC1

// Load pointer to IV (No need to write length (CSS\_DMA\_SRC1 works by default with 16 bytes length)

cssv2\_load\_dma\_src1(cssv2\_context, IV\_PTR);

[W32: A=50190028 W=30000824](#_CSS_DMA_SRC1)



##### KEY 地址写入SRC2

// Load pointer to KEY

cssv2\_load\_dma\_src2(cssv2\_context, KEY\_PTR);

cssv2\_load\_dma\_src2\_len(cssv2\_context, KEY\_SIZE);

[W32: A=50190030 W=30000834](#_CSS_DMA_SRC2)

[W32: A=50190034 W=00000010](#_CSS_DMA_SRC2_LEN)



##### 运算返回数据地址写入RES0

// Load pointers for final result (No need to write length (CIPHER command sets CSS\_DMA\_RES automatically to 1 AES block, 16 bytes)

cssv2\_load\_dma\_res0(cssv2\_context, RES\_PTR);

[W32: A=50190038 W=30000844](#_DMA_CSS_RES0)

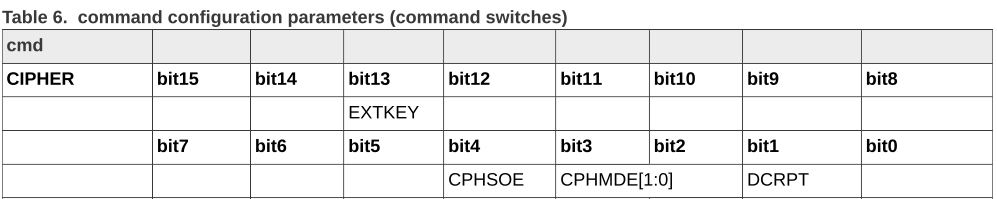


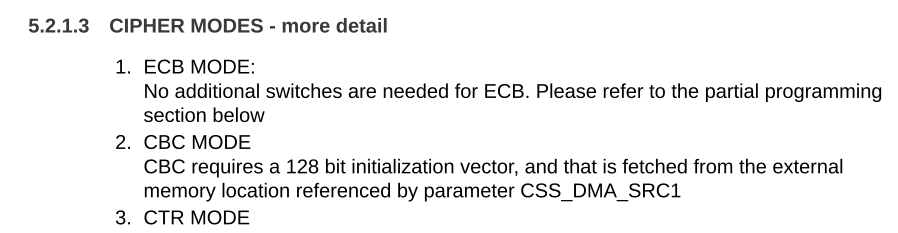
#### 配置command

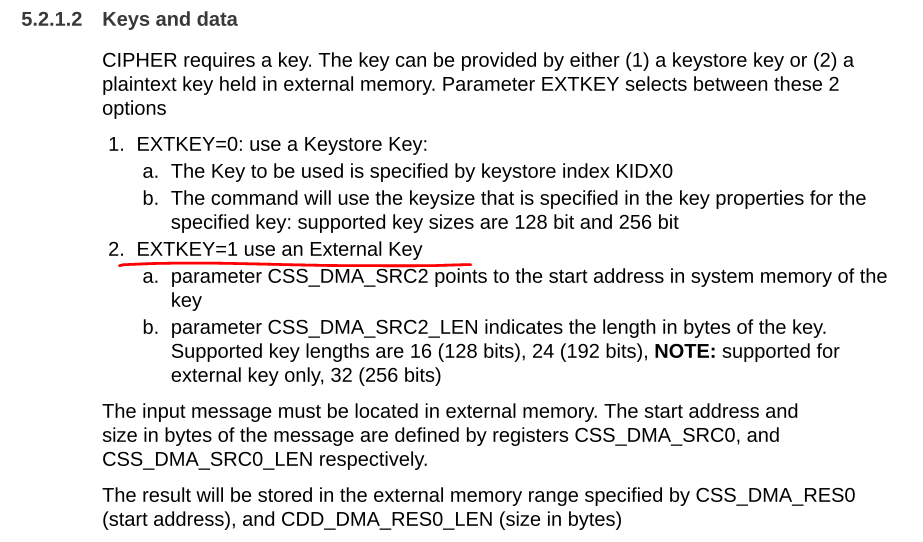
配置CBC mode， 选择external key。

// Write configuration word

cssv2\_write\_cmd\_cfg(cssv2\_context, cmd\_cfg\_word[tv\_idx]);

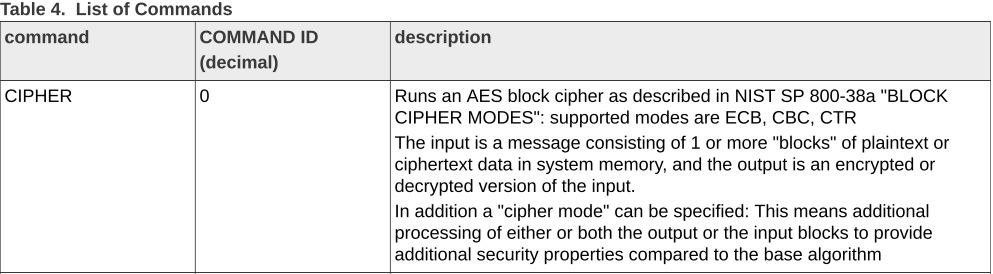
[W32: A=50190008 W=00002004](#_CSS_CTRL（enable/start）)





#### start CSS

command ID 设置为0，启动CSS



// Start cipher operation

cssv2\_start\_op(cssv2\_context, CSS\_CMD\_CTRL\_CIPHER);

// -------------------------------------------------------------------------

// -- Wait for CSS process completion

cssv2\_check\_int(cssv2\_context);

[W32: A=50190004 W=00000003](#_CSS_CTRL（enable/start）)

#### 检验DMA 返回的加解密数据

// -------------------------------------------------------------------------

// -- Check results

// -------------------------------------------------------------------------

// Ciphertext

for (mem\_idx = 0; mem\_idx < (RES\_SIZE / 4); mem\_idx++) {

R32(REG32(RES\_PTR + (mem\_idx << 2)), rdata);

if (rdata != ciphertext[tv\_idx][mem\_idx]) {

SHOW32(STIMNAME," MISMATCH: Data[%d] Mismatch!!!", mem\_idx);

SHOW32(STIMNAME," Exp: 0x%08x", ciphertext[tv\_idx][mem\_idx]);

SHOW32(STIMNAME," Act: 0x%08x", rdata);

ERROR(STIMNAME, "Fail : Ciphertext is incorrect value");

}

else {

SHOW32(STIMNAME," MATCH: Data[%d] Match!!!", mem\_idx);

SHOW32(STIMNAME," Exp: 0x%08x", ciphertext[tv\_idx][mem\_idx]);

SHOW32(STIMNAME," Act: 0x%08x", rdata);

INFO(STIMNAME, "Pass : Ciphertext is correct value");

}

}

//

}

## Case Porting

CORE\_S3\_RESOURCES

根据这个宏判定MU是soc端的还是S200端的。

用到的mem map定义

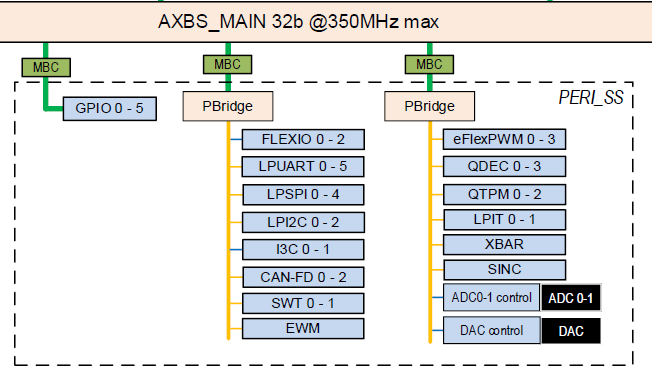
/home/imx95\_verif/usr/nxa28190/imx943\_1p45/testbench/common\_blocks/v\_ss\_imx943\_cssi\_vip/tool\_data/sray/mem\_map

# EWM

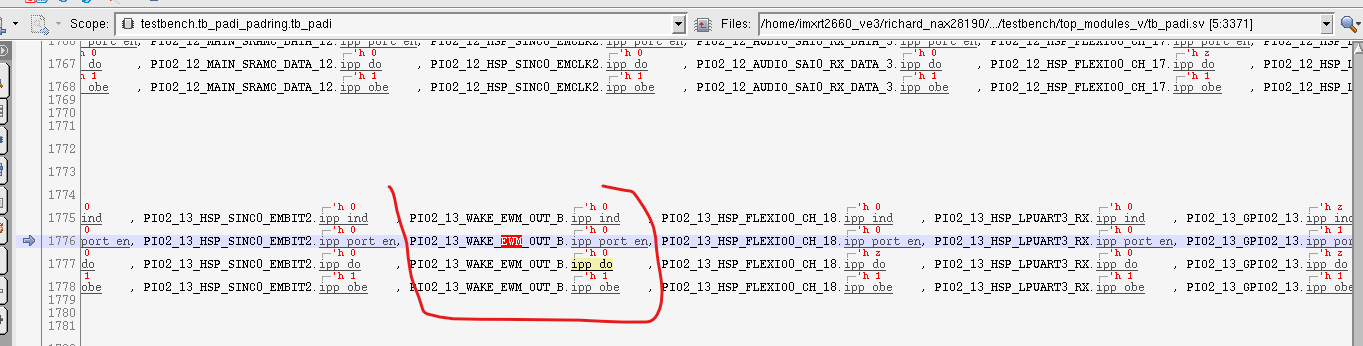
## Issue

[TKT0663359](javascript:emxFormLinkClick(%22../common/emxTree.jsp?relId=null&emxSuiteDirectory=components&mode=View&jsTreeID=undefined&otherTollbarParams=relId%2CemxSuiteDirectory%2Cmode%2CjsTreeID&objectId=11392.59601.54220.56479&relId=null%22,%20%22popup%22,%20%22%22,%20%22%22,%20%22%22,%20%22TKT0663359%22,%20%22%22,%20%22%22))  [RT2660]EWM LPO clock was tie 0

**D\_IP\_EWM\_SYN\_1.19**



## PIN\_MUX



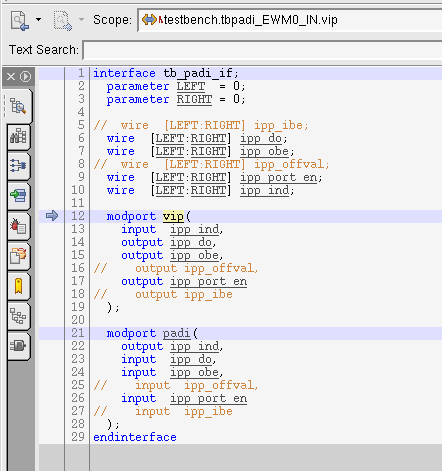
Ipp\_port\_en =1, IPP\_IND 表示EWM 向GPIO输入信号

EWM 只有输出信号。  
如果ipp\_port\_en=1/ipp\_obe=1. Ipp\_do表示GPIO向peripheral输出信号

### VIP

##### Interface ewm\_out/ewm\_in

testbench/common\_blocks/v\_ip\_tbpadi/testbench/modules\_v/4\_wire/tb\_padi\_if.sv



##### 例化

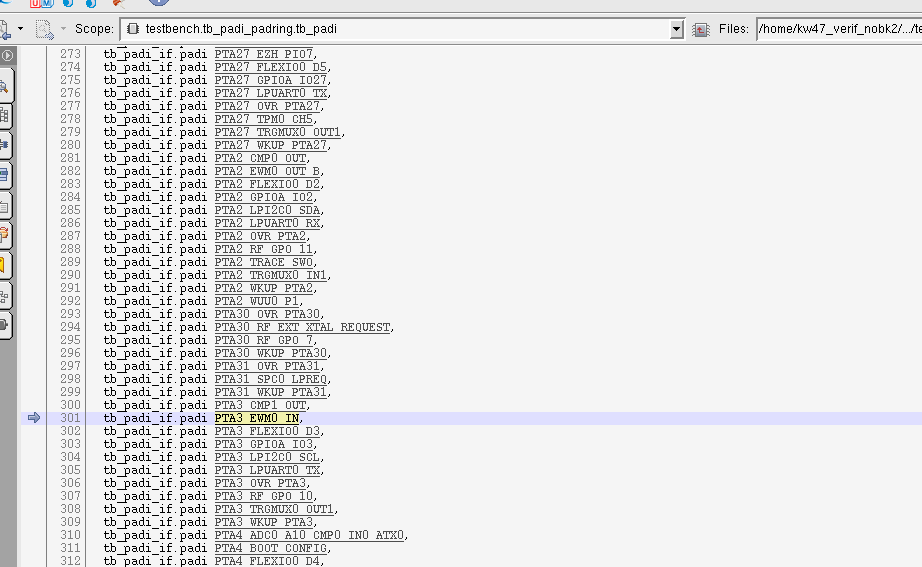


### MUX PTA3\_EWM0\_IN

tbpadi\_EWM0\_IN 的信号链接到PTA3\_EWM0\_IN

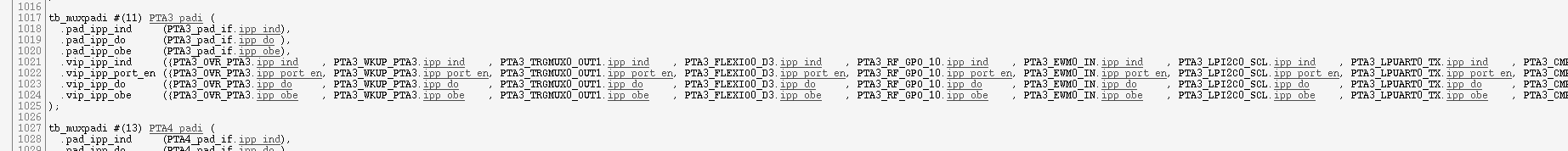
将VIP中的ipp\_ind连接上

[PTA3\_EWM0\_IN.ipp\_port\_en](#_V的配置)



#### C的配置

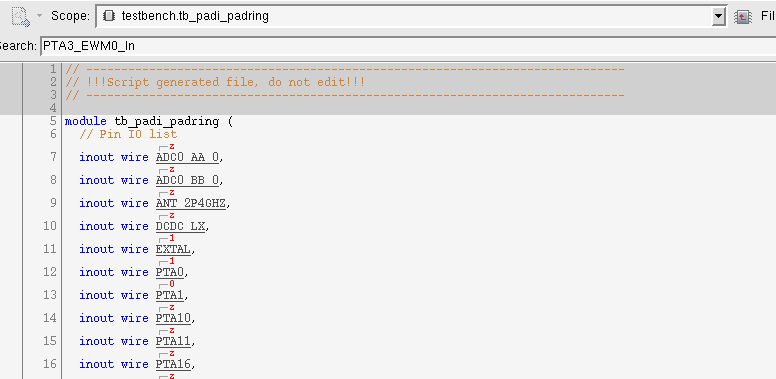
#### V的配置



### tb\_padi\_padring

##### 定义

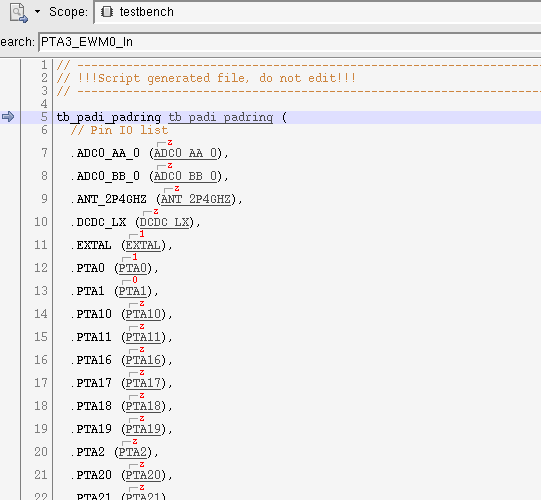
V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_modules\_v/tb\_padi\_padring.sv

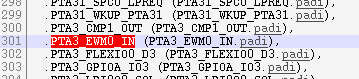




##### 例化

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_instances\_v/tb\_padi\_padring\_inst.sv





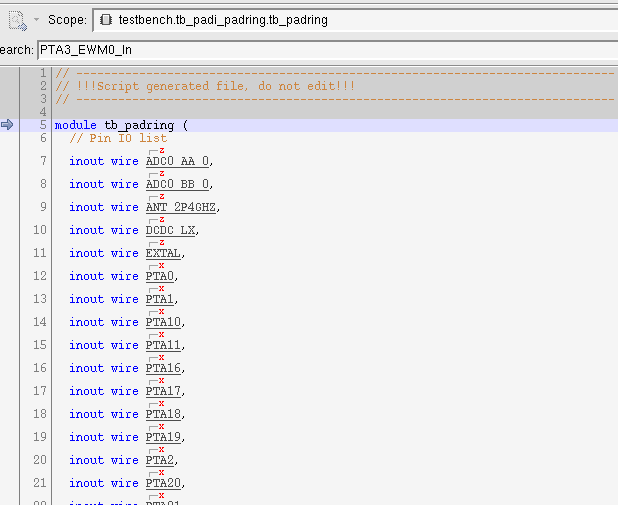
#### Tb\_padring

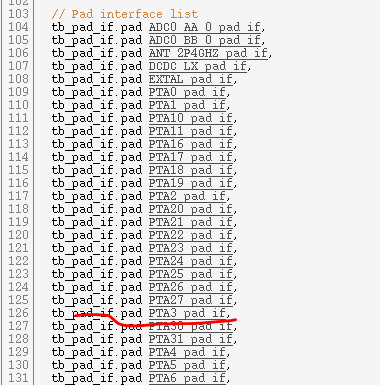
将PTA3 口连接到interface PTA3\_pad\_if.pad上

1. 从外面传输数据给PAD， VIP端口需要enable obe&port\_en. PAD接受do信号。
2. PAD传送数据给外面：ind接收PAD数据

##### 定义

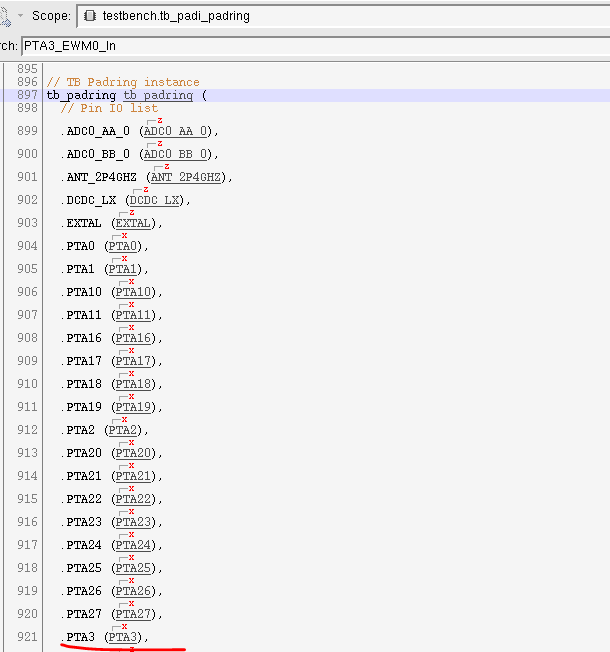
V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_modules\_v/tb\_padring.sv

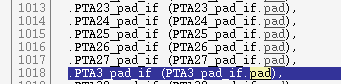




##### 例化

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_modules\_v/tb\_padi\_padring.sv





##### 链接

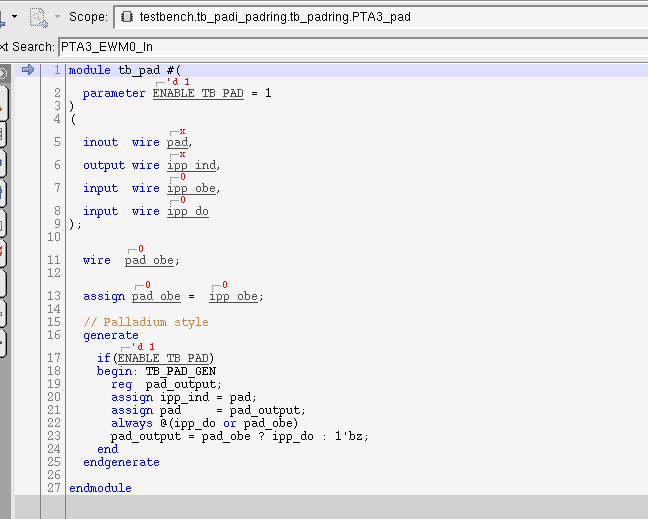
###### 定义

芯片PAD 接到 [PTA3\_pad\_if](#_Tb_pad_if)

如果从外面传输数据给PAD， VIP端口需要enable obe&port\_en. PAD接受do信号。

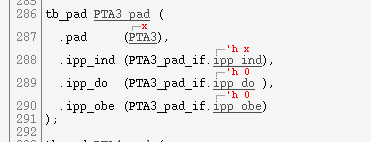
PAD传送数据给外面：ind接收PAD数据

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/common\_blocks/v\_ip\_tbpadi/testbench/modules\_v/4\_wire/tb\_pad.sv



###### 例化

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_modules\_v/tb\_padring.sv

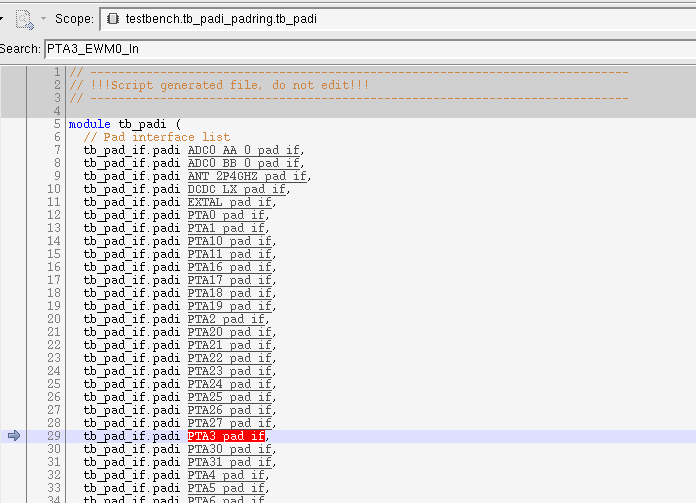


#### Tb\_padi

[PTA3\_pad\_if.padi](#_Tb_pad_if) 连接到[PTA3\_EWM0\_IN.padi](#_Tb_padi_if)上

##### 定义

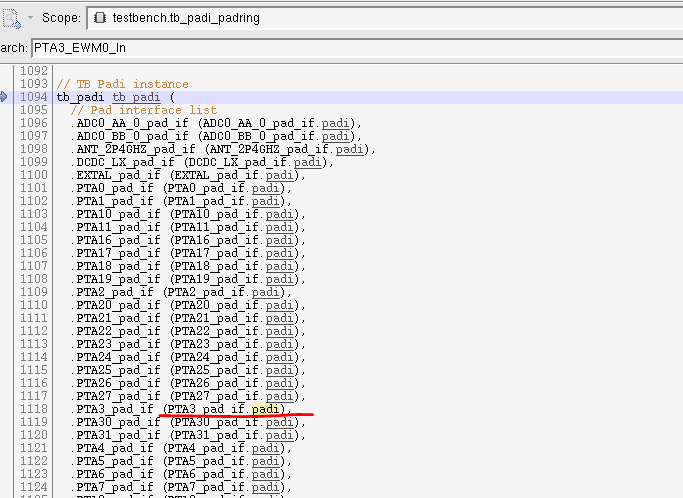
V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_modules\_v/tb\_padi.sv

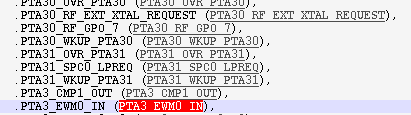




##### 例化

： V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_modules\_v/tb\_padi\_padring.sv





##### 链接

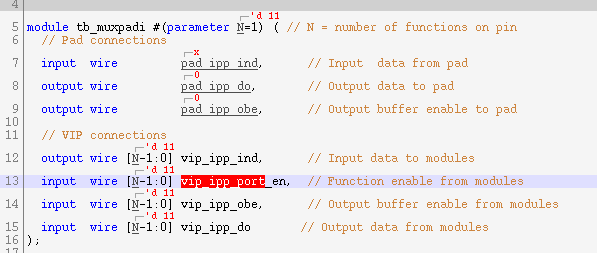
跟据port\_en信号，选中PAD的ALT 模式。

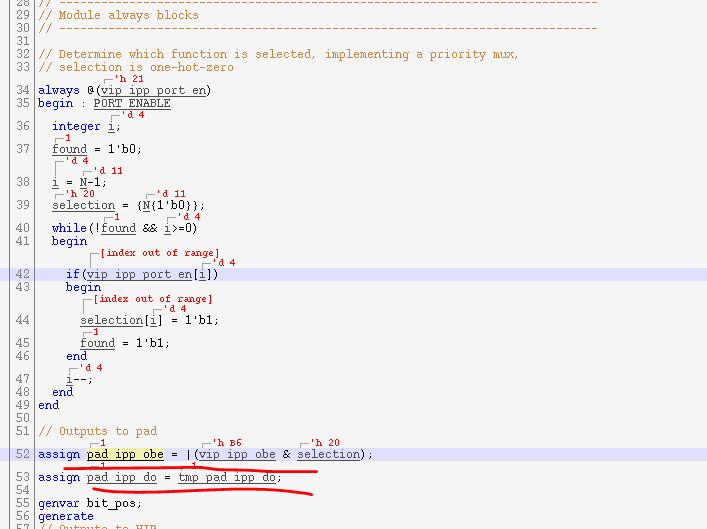
如果有多个enable信号，选中最大的那个

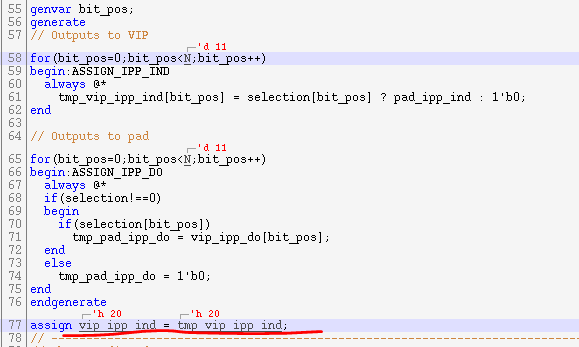
这里定义了PAD到VIP的信号传输Tb\_pad\_if.padi 到 tb\_padi\_if.padi 的传输

###### 定义

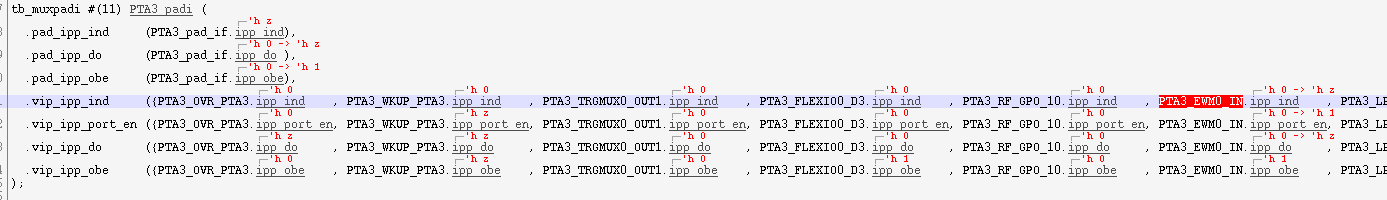
V\_SS\_KW47\_SOC\_TB\_1.37/testbench/common\_blocks/v\_ip\_tbpadi/testbench/modules\_v/4\_wire/tb\_muxpadi.sv







###### 例化



### Tb\_padi\_mux3

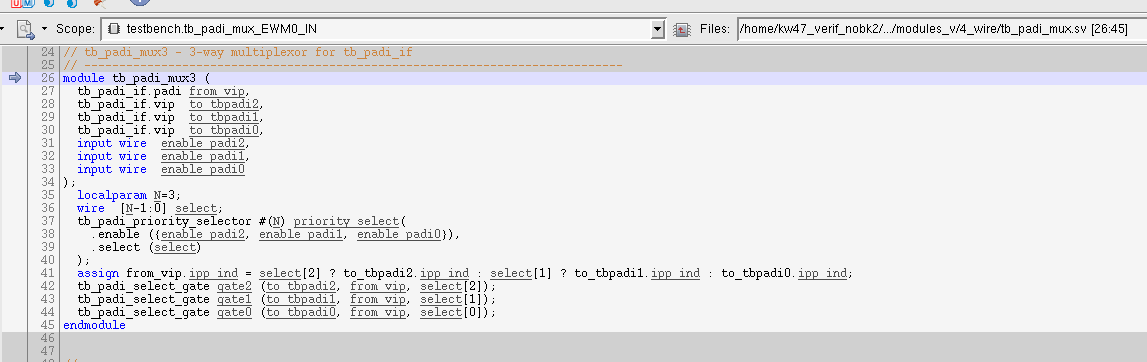
PTA3\_EWM0\_IN.vip 接到tbpadi\_EWM0\_IN.padi（VIP interface）

EWM0\_IN支持三个引脚。 PTA3/PTA17/PTA20

通过MUX选择对应的引脚PTA3\_EWM0\_IN.vip/PTA17\_EWM0\_IN/PTA20\_EWM0\_IN

#### 定义

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/common\_blocks/v\_ip\_tbpadi/testbench/modules\_v/4\_wire/ tb\_padi\_mux.sv



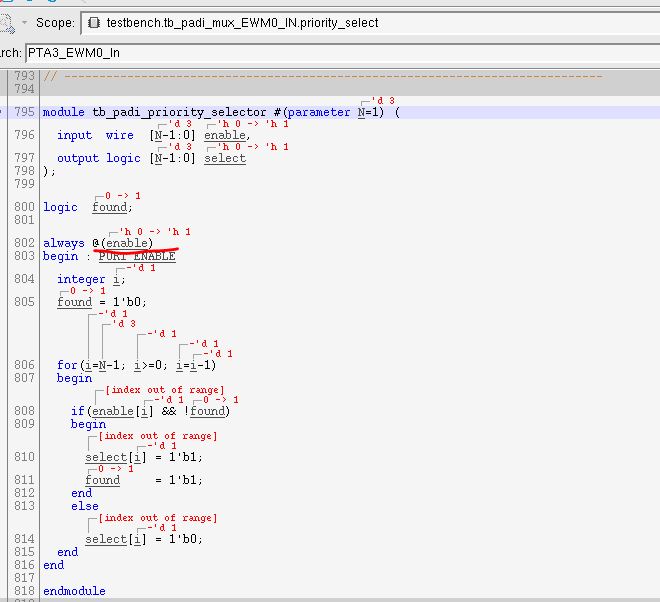
##### 根据priority进行select

C进行soc\_allocate\_ip后，相关的PIN寄存器会被选中配置。

SV根据RTL中的选中pin，进行[Port enable](#_V)

该FUNCTION 根据port enable信号，进行相关PADI MUX的选择。

优先级从大到小，比如同时开两个enable，这里只会选择最大的那个PADI2>PADI1>PADI0



###### Assign Ipp\_ind



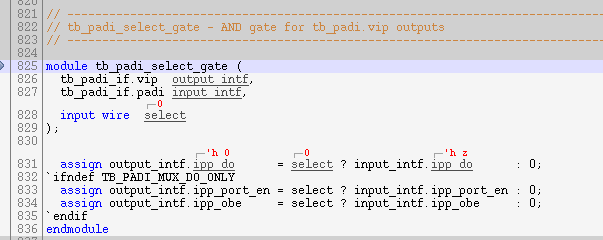
Mux上的输入，接到VIP的输入

ASSIGN tbpadi\_EWM0\_IN.padi.ipp\_ind = PTA3\_EWM0\_IN.vip.ind

###### Assign ipp\_do

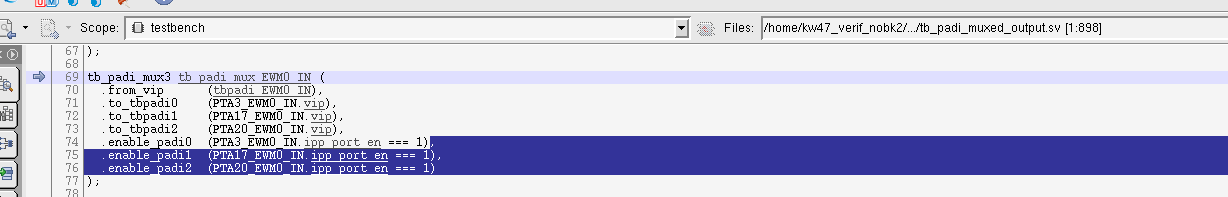
根据select assign ipp\_do。VIP 口的输出接到mux上的输入

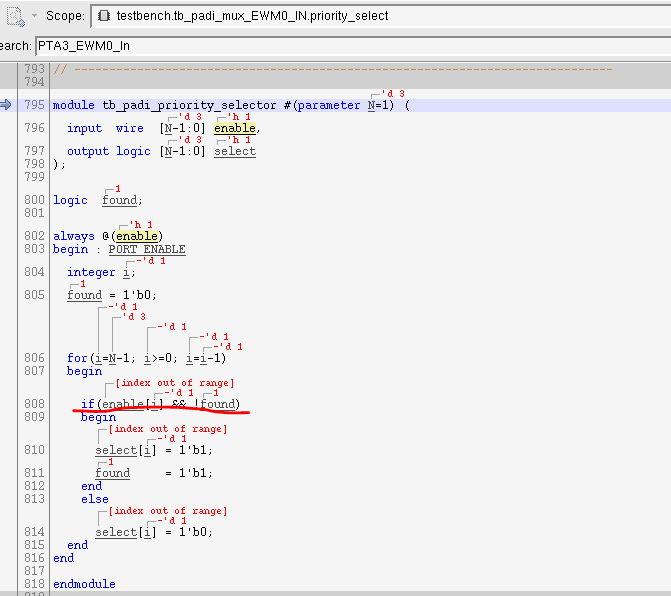
ASSIGN tbpadi\_EWM0\_IN.padi.ipp\_do = PTA3\_EWM0\_IN.vip.ipp\_do



#### 例化

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_instances\_v/tb\_padi\_muxed\_output.sv





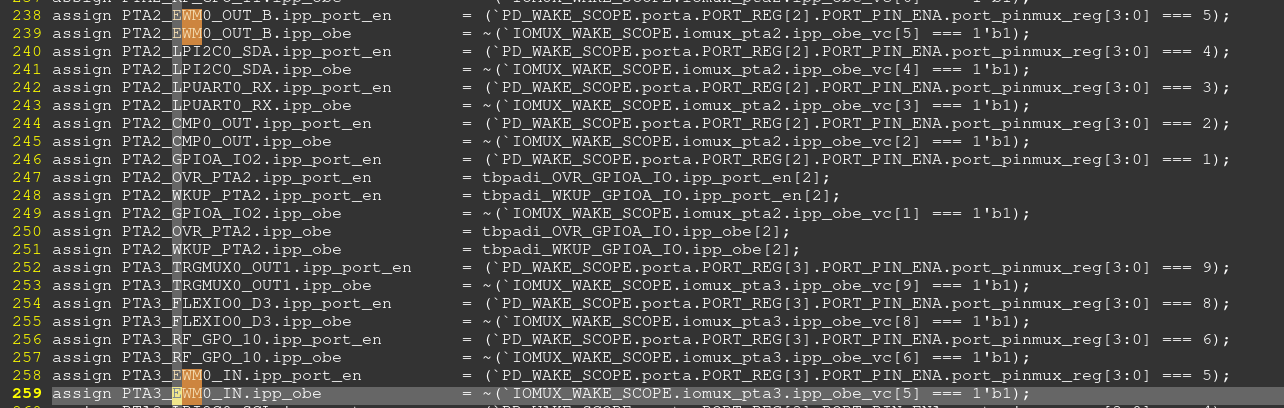
#### 链接

##### V

当C中的配置完成后，SV会根据RTL中的信号选择使能的pin。

PTA3\_EWM0\_IN.ipp\_port\_en 置1. Tb\_padi\_mux3会将VIP 的interface 连接到PTA3\_EWM0\_IN

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_wires\_v/tb\_padi\_assigns.sv



##### C

执行如下代码后

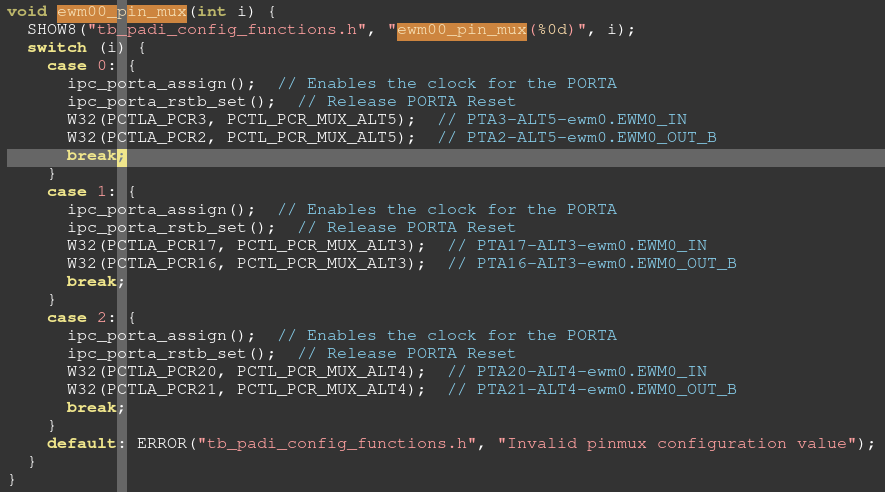
soc\_new\_request(&ip\_request, EWM\_IP);

soc\_allocate\_ip(&ip\_request);

会根据arg中的参数进行pinmux的配置。 这里选择了PTA3，进行相关寄存器配置。

CORE\_A\_C\_ARG += EWM\_PINMUX\_OPT=PIN\_OPT0

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/tool\_data/compiler/include/tb\_padi\_config\_functions.h



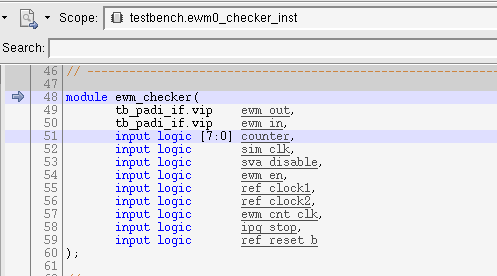
### VIP

传入的interface 为tbpadi\_EWM0\_IN.vip

#### Ewm\_checker模块定义

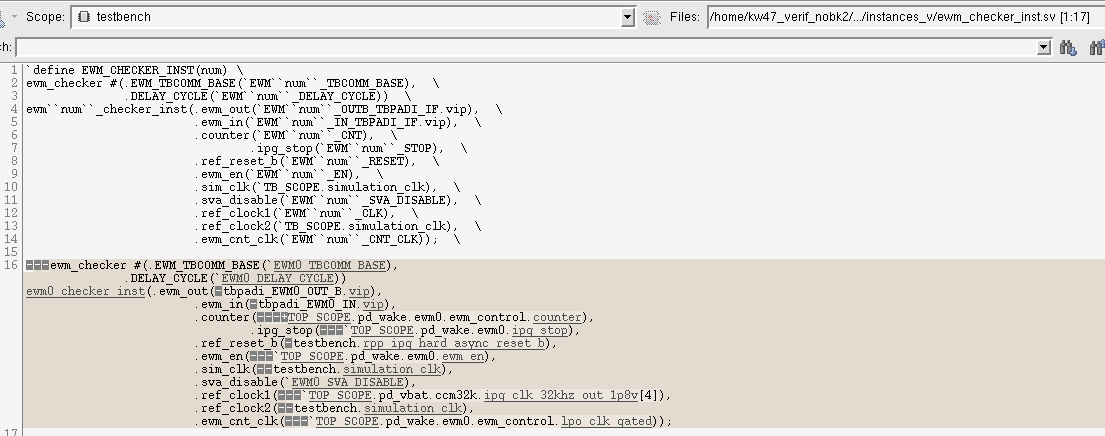
模块定义ewm\_checker:

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/common\_blocks/v\_ms\_ewm\_nxt\_vip/testbench/modules\_v/ewm\_checker.sv



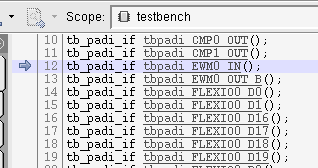
#### 模块例化ewm0\_checker\_inst

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/common\_blocks/v\_ms\_ewm\_nxt\_vip/testbench/instances\_v/ ewm\_checker\_inst.sv

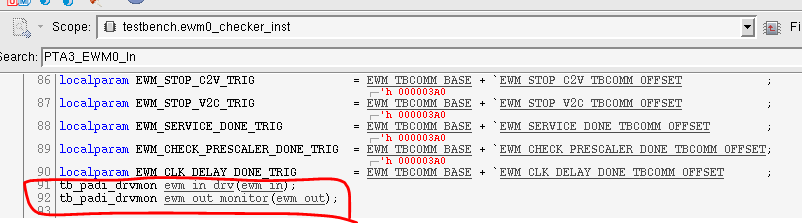


##### 传入的interface tbpadi\_EWM0\_IN

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/blocks/soc\_tb/testbench/top\_wires\_v/ tb\_padi\_muxed\_output\_if\_inst.sv



#### 驱动pad



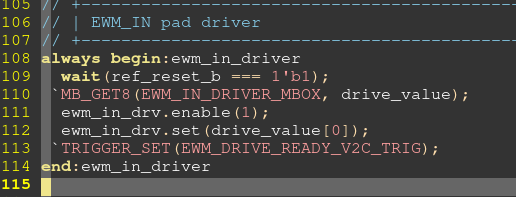
pad驱动tb\_padi\_drvmon ewm\_in\_drv

例化两个驱动monitor，传入的参数为interface

用来控制pad的传输

V\_SS\_KW47\_SOC\_TB\_1.37/testbench/common\_blocks/v\_cs\_pin\_clk\_drvmon/testbench/modules\_v/4\_wire/ tb\_padi\_drvmon.sv

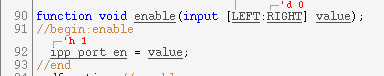
##### ewm\_in\_drv



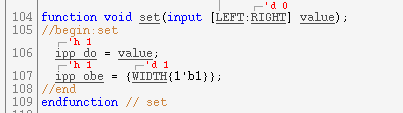
###### 定义



###### Enable



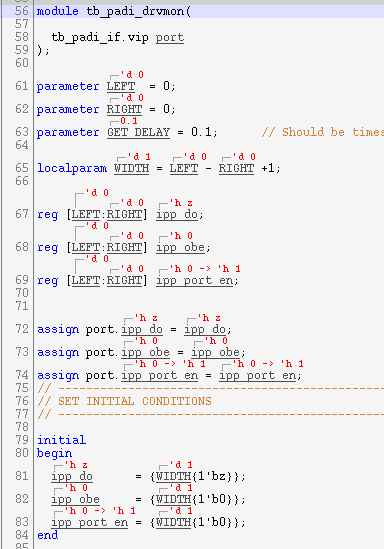
###### Set



##### ewm\_out\_monitor



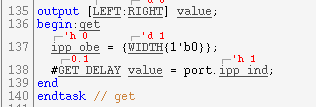
###### 定义



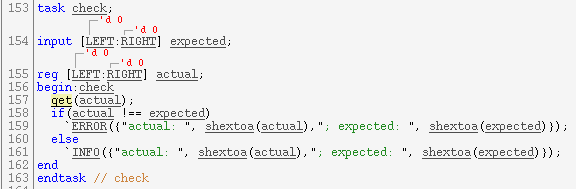
###### Enable



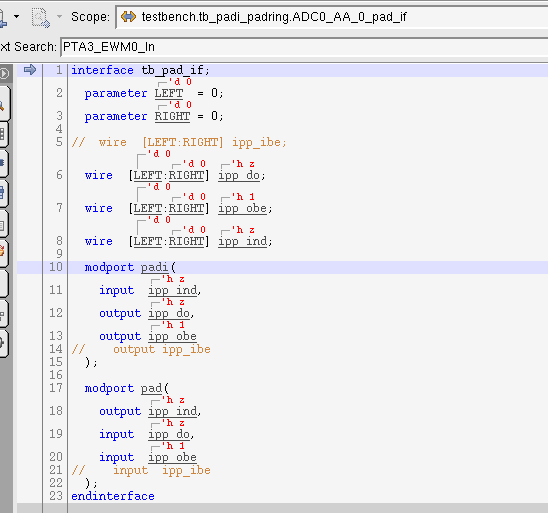
###### Get



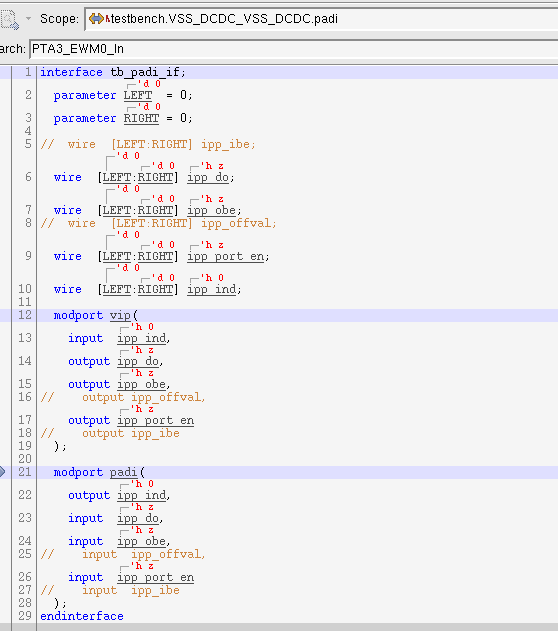
###### Check



### Tb\_pad\_if

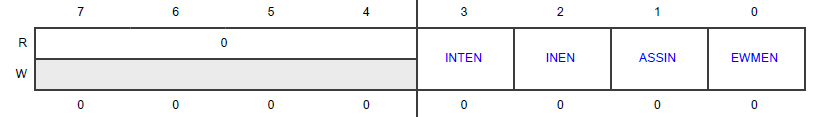


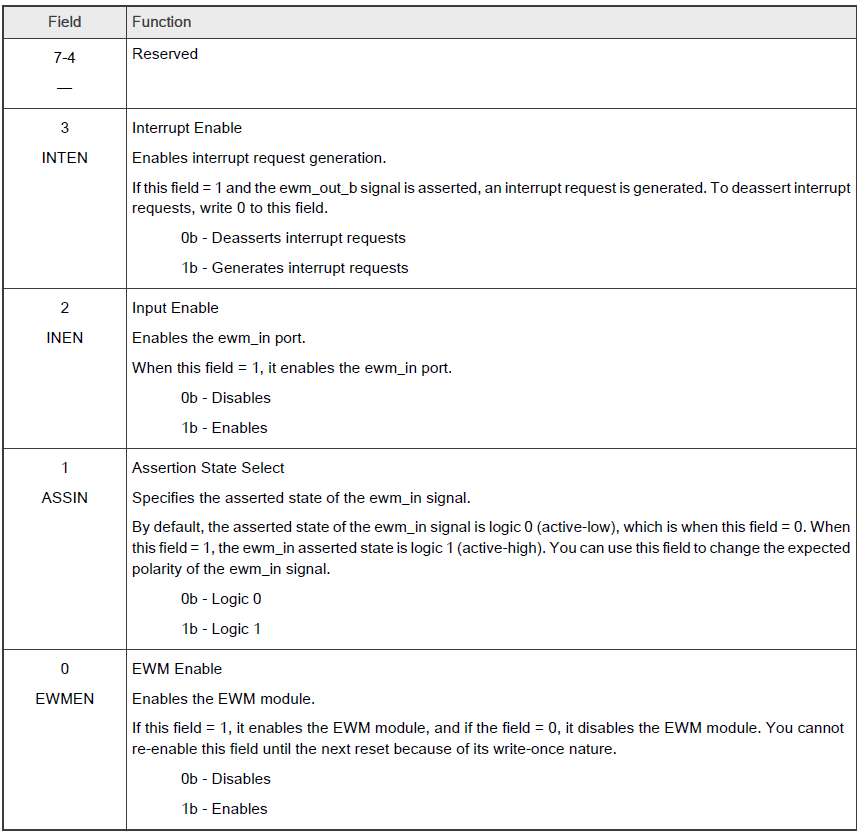
### Tb\_padi\_if



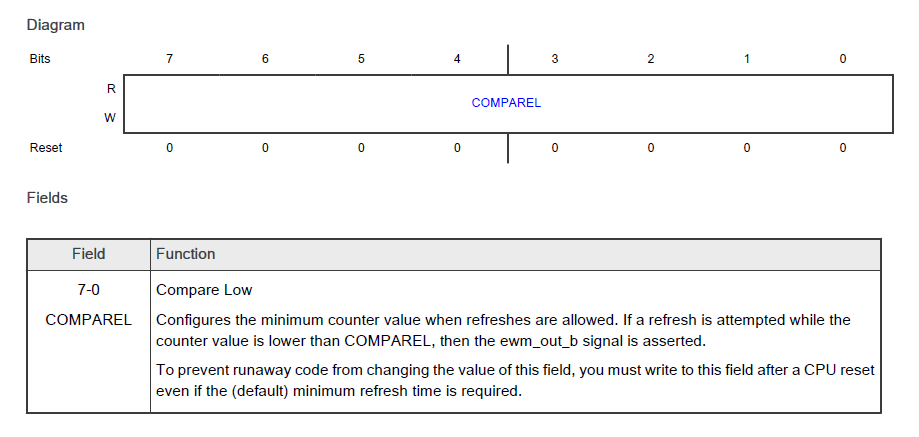
## Register

### CTRL

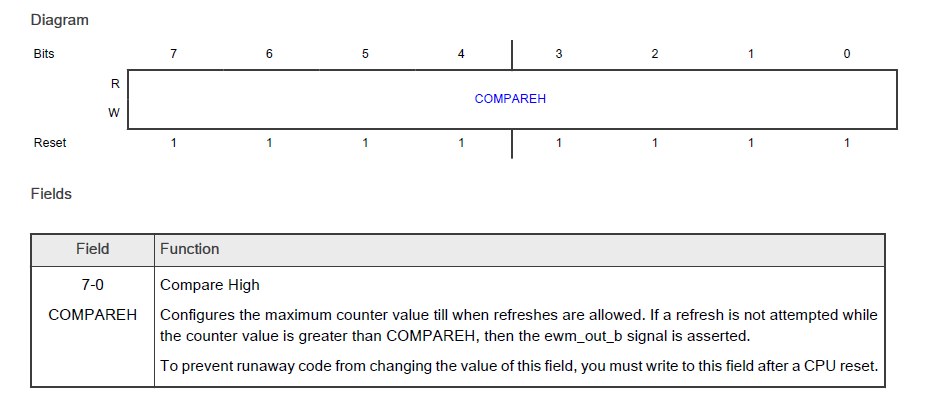




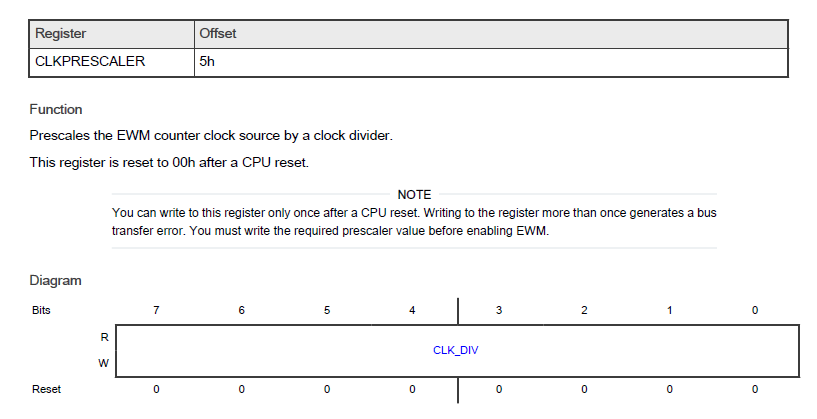
### COMPARE LOW

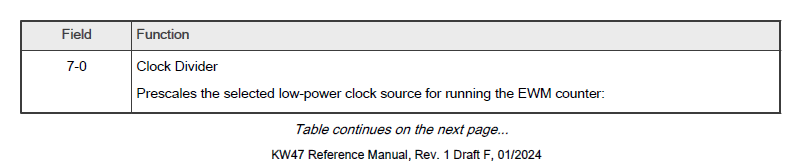


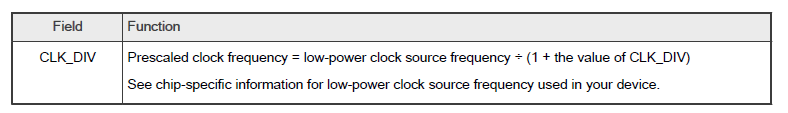
### COMPARE HIGH



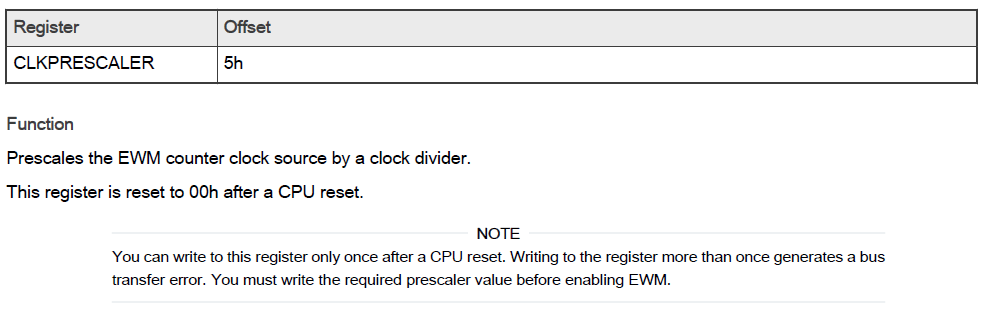
#### CLKPRESCALER

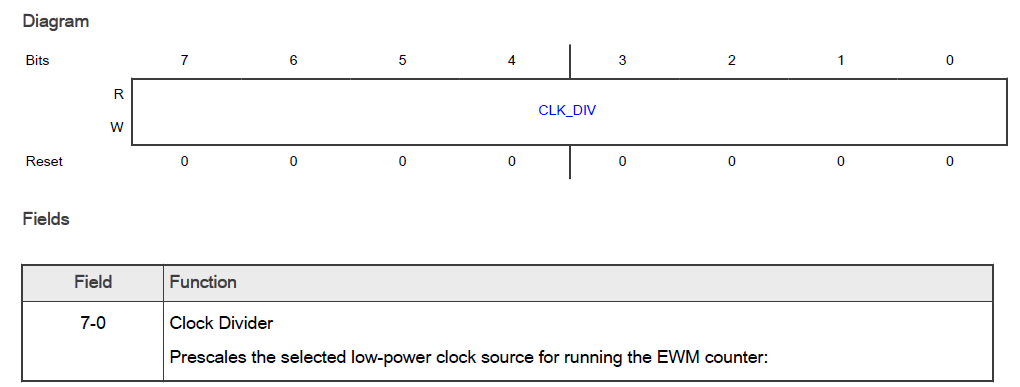


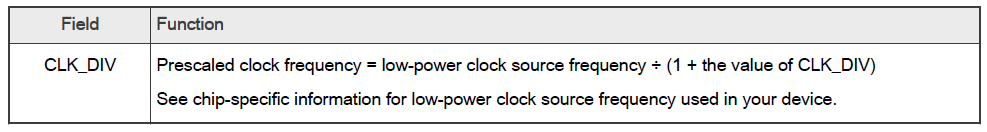




### CLKPRESCALER







## Function

### Ewm\_out\_b

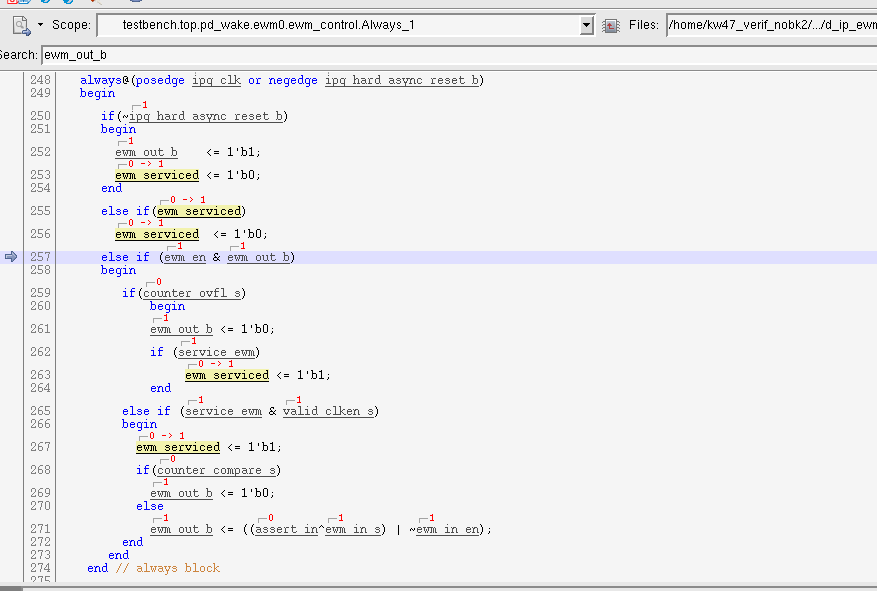
默认为1，0表示assert。

如果设置了CMPH/CMPL，EWM enable后

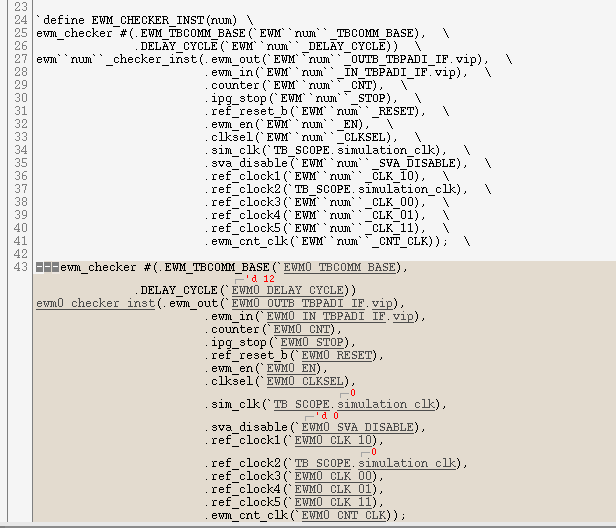
* ASSIN = 0（默认）
  + 等待counter自动计数（LPO clock），counter >CMPH（overflow），不管EWM\_in， ewm\_out自动assert
  + 如果counter<CMPL的时候refresh EWM，EWM\_OUT 自动assert
  + 如果CMPL<counter <CMPH的时候refresh EWM， EWM\_OUT是否assert需要看EWM\_IN.

如果ASSIN=0时，设置EWM\_IN =0(表示assert)，EWM\_OUT会assert。设置EWM\_IN=1(不assert)，EWM\_OUT不会assert。 Refresh后如果EWM\_OUT 没有assert，counter会重新计数

* ASSIN =1(EWM\_IN=1 时表示assert状态)
  + Overflow， counter<CMPL的情况与如上一致。
  + 只有CMPL<counter<CMPH时，EWM\_IN=1, EWM\_OUT会assert。EWM\_IN=0时，EWM\_OUT不会assert。Refresh后，如果没有assert，counter重新计数



## 2660 ewm vip



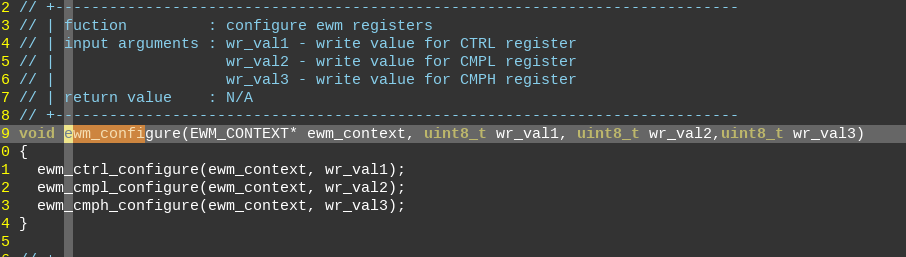
## API

### ewm\_configure

第一个参数配置ctrl寄存器

第二个参数配置CMPL 寄存器

第三个参数配置CMPH寄存器



### Ewm\_check\_counter

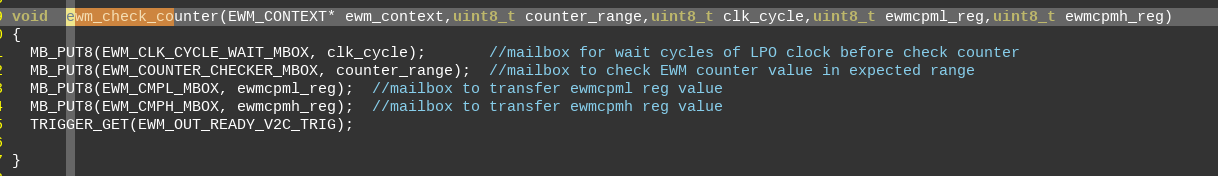
使用GPO[2] clock计数，注意C中要配置选择GPO2为计数时钟

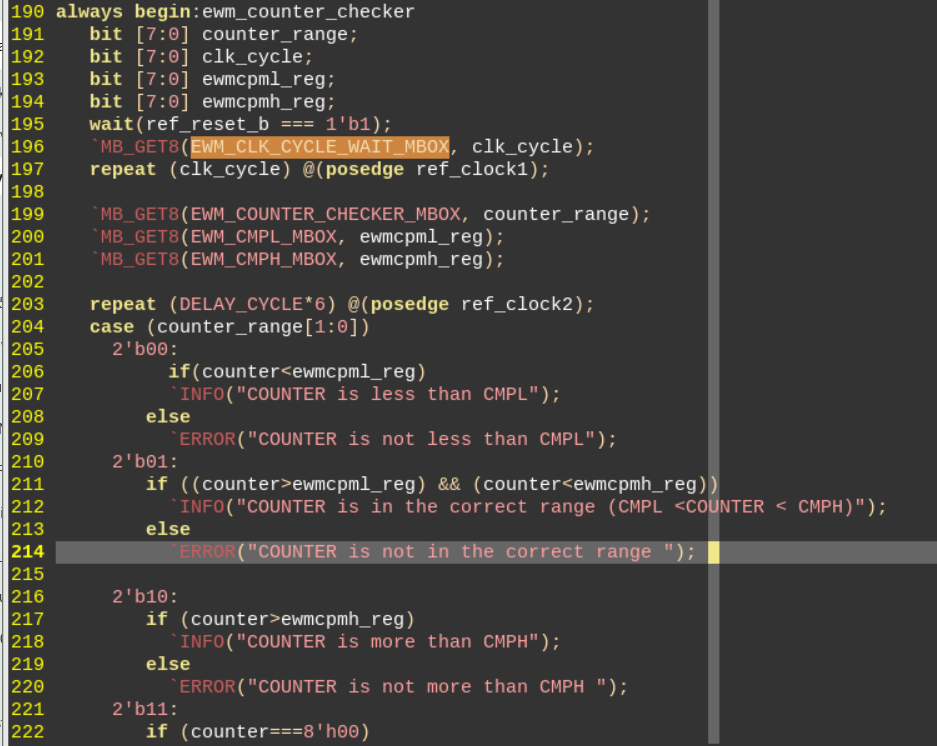
第一个参数：counter\_range

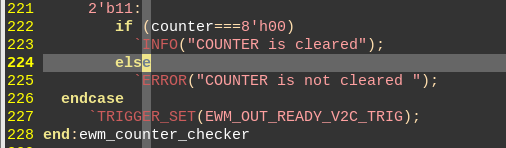
0: SV检测是此时EWM counter值是否小于CMPL  
1: SV检测EWM counter是否位于CMPL -CMPH之间  
2. 检测EWM counter是否大于CMPH  
3.检测EWM counter值是否为0

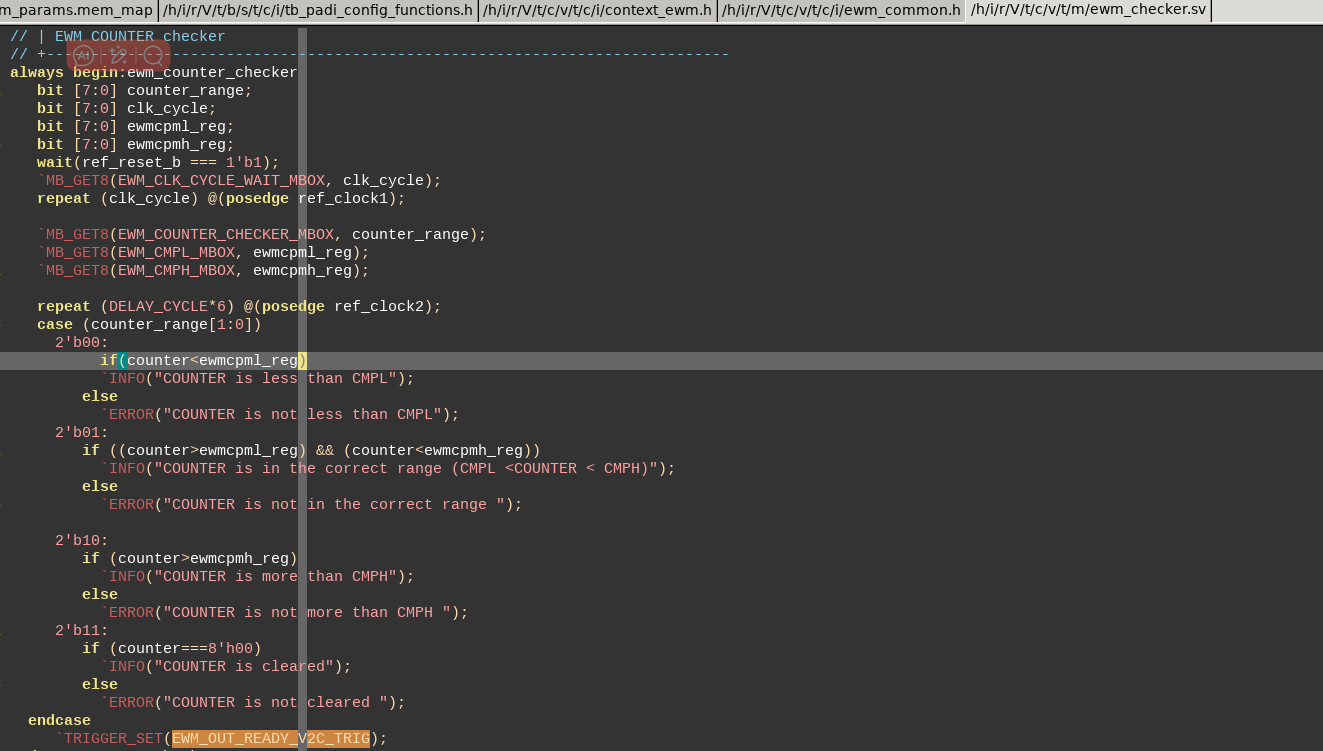
第二个参数：clk\_cycle

表示在SV进行第一个参数的操作前，delay 几个（计数counter用的）clock







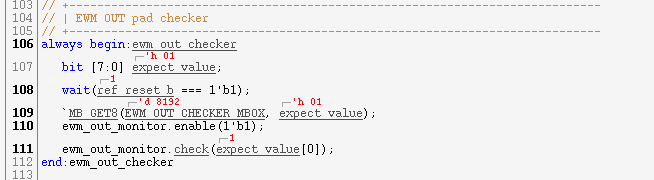


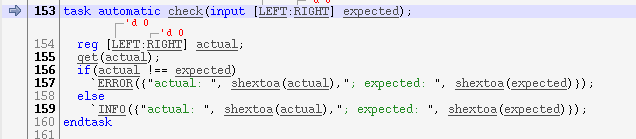
### Ewm\_check\_output

C:



SV:





# SWT0

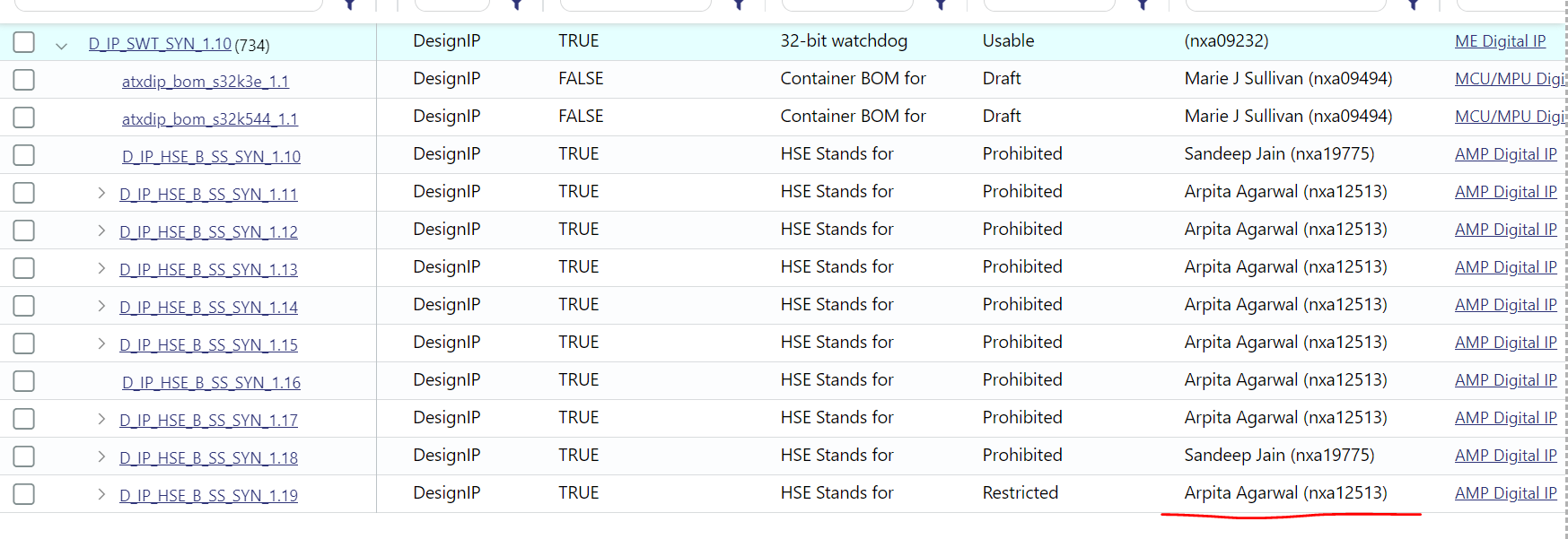
D\_IP\_SWT\_SYN\_1.10

Refer： S32K5

## Contact

印度服务器awsc1.  
/home/r43\_verif/users/michael

IP design: Nancy Amedeo (nxa09423)

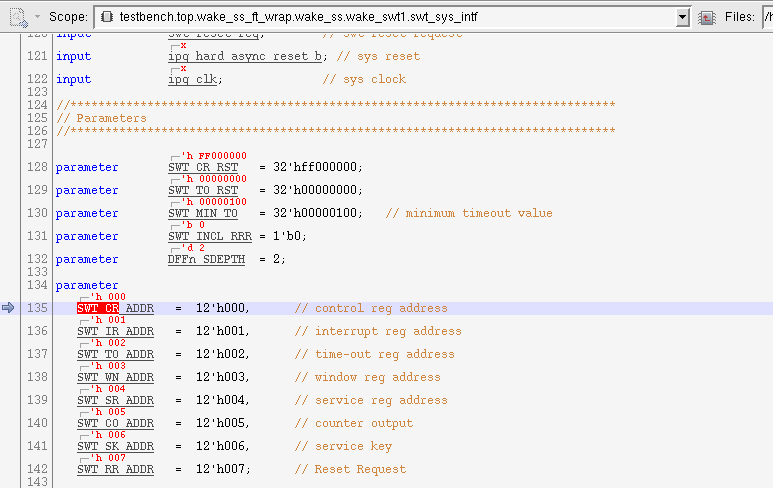


VIP： V\_MS\_SWT\_STIM

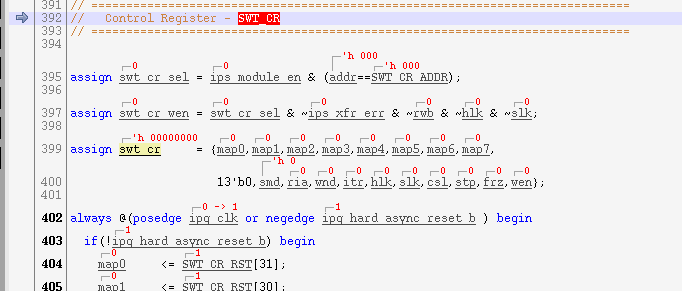
Abhinav Gaur (nxa11562) / Tarun Gupta (nxf91281)

## RTL

### RESET VALUE



### SWT\_CR



# Low power

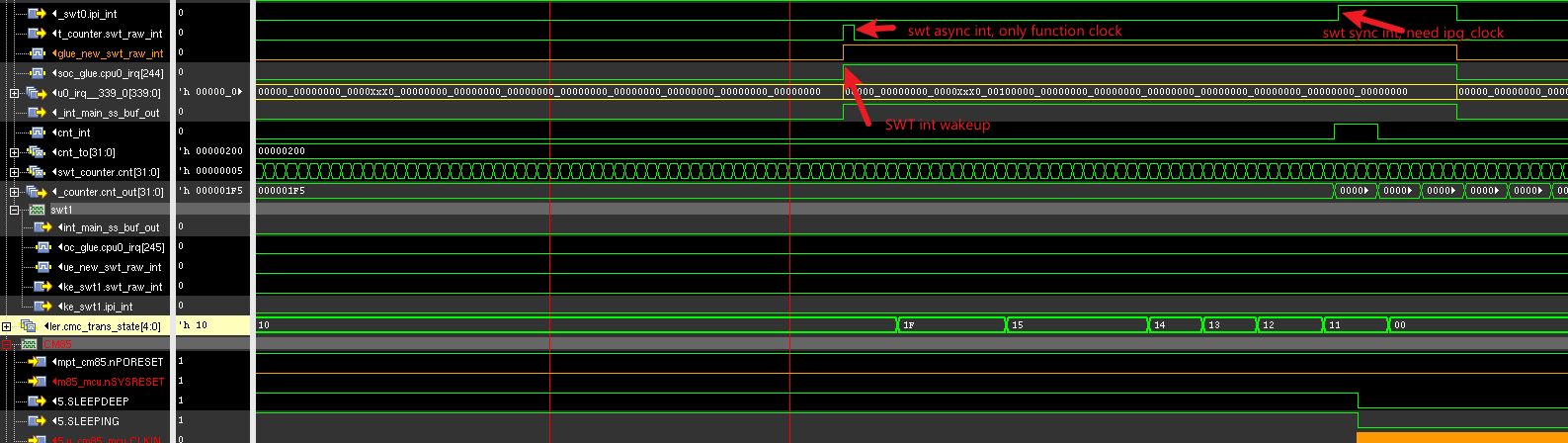
## CMC 状态寄存器

testbench.top.syscon\_ss\_ft\_wrap.syscon\_ss.syscon\_powercon.rt2660\_powercon.CMC\_INST[0].cmc\_controller.cmc\_trans\_state[4:0]

进入sleep/deepsleep/power down/deep power down CMC 的状态都是10

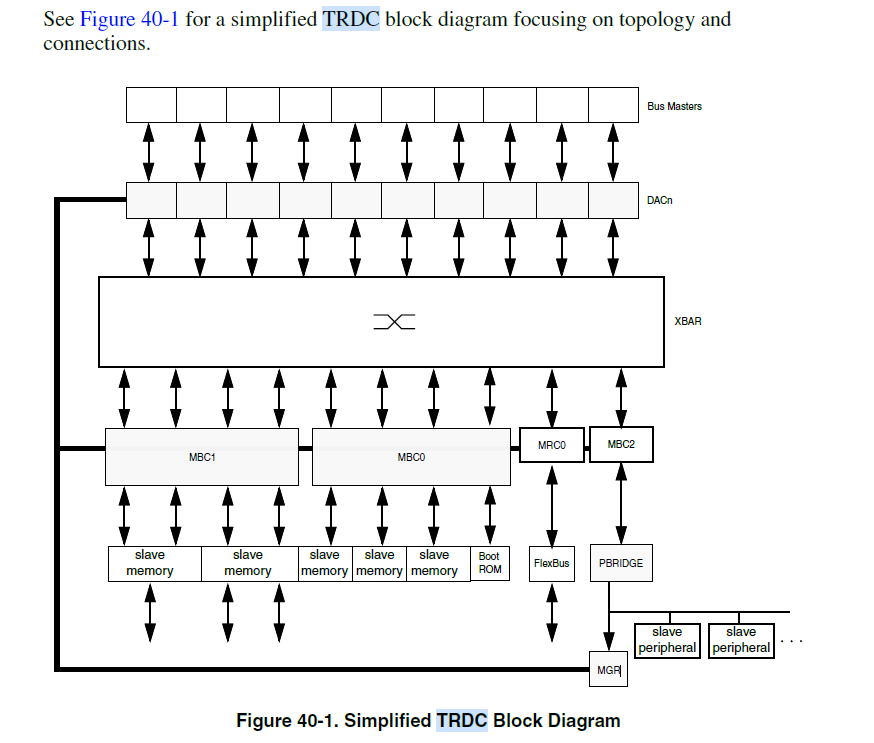
1F 是开始进入唤醒程序



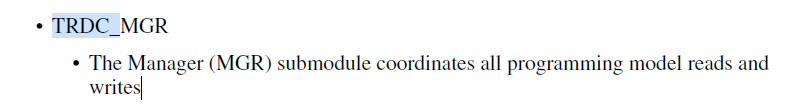


# TRDC

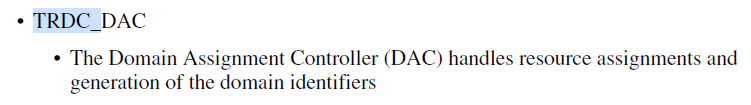
1180 为例：



## MGR



## DAC



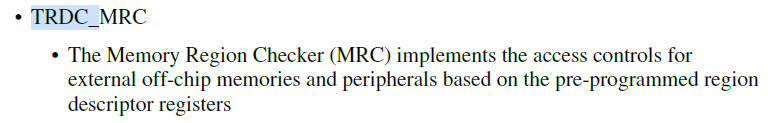
DAC 用于给bus master生成domain identifier。

对用的配置寄存器为：

## MBC



## MRC



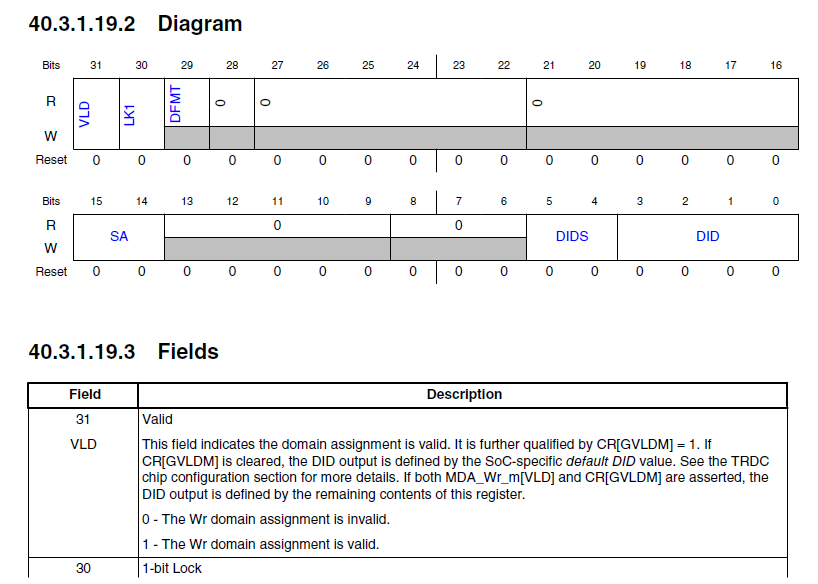
## Register

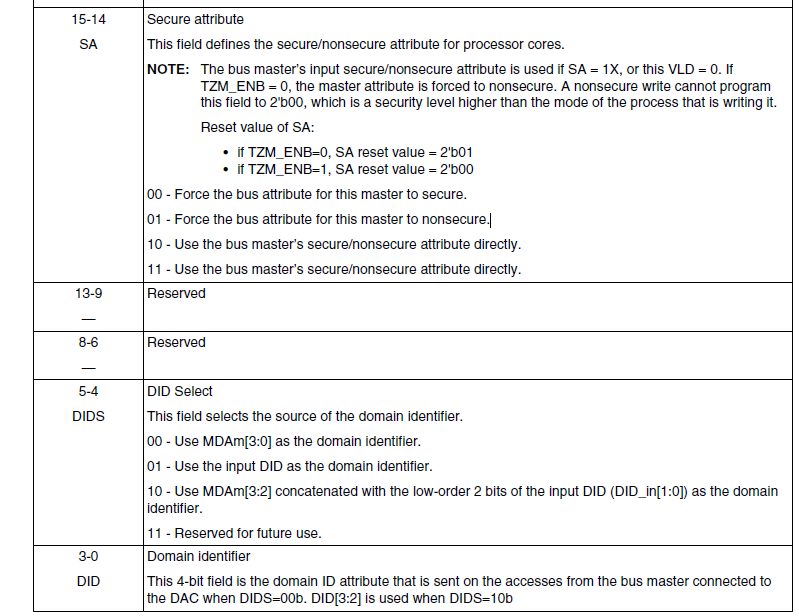
### DAC master domain assignment

Bit31用于enable DAC

Bit15-14 ： secure、privilege配置







# S110

## Preload

/home/imxrt2660\_ve3/richard\_nax28190/V\_SS\_RT2660\_SOC\_TB\_1.15/testbench/blocks/soc\_tb/testbench/instances\_v/s110\_rom\_init.sv

### RT2660\_MEM\_LOADER

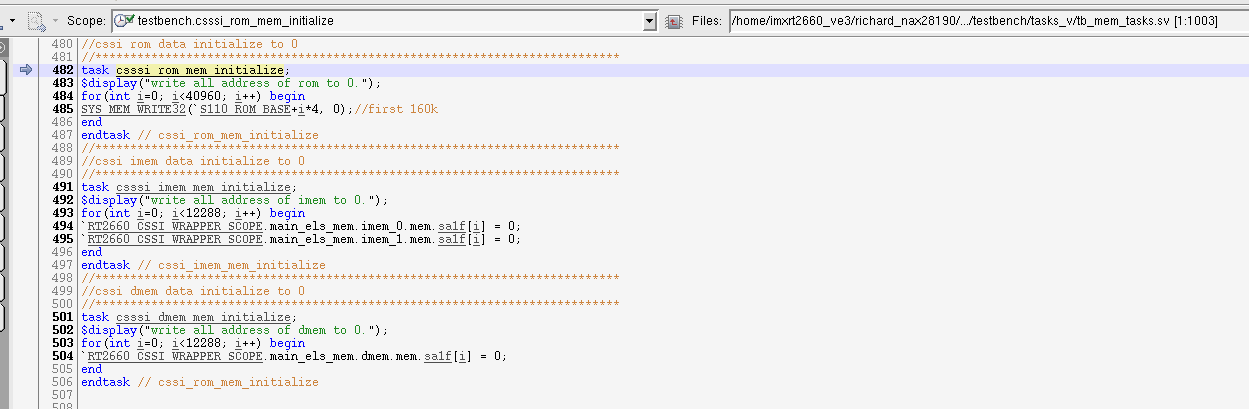
Top\_tasks\_v/tb\_mem\_loader.sv

1. 初始化CSSI ROM,IMEM,DMEM, IMEM/DMEM的[初始化](#_Mem_initialize)是向sa1f中写入数据,
2. load hex到memory中，
3. IMEM/DMEM SA1F中数据swap到mem中。



### Mem\_initialize

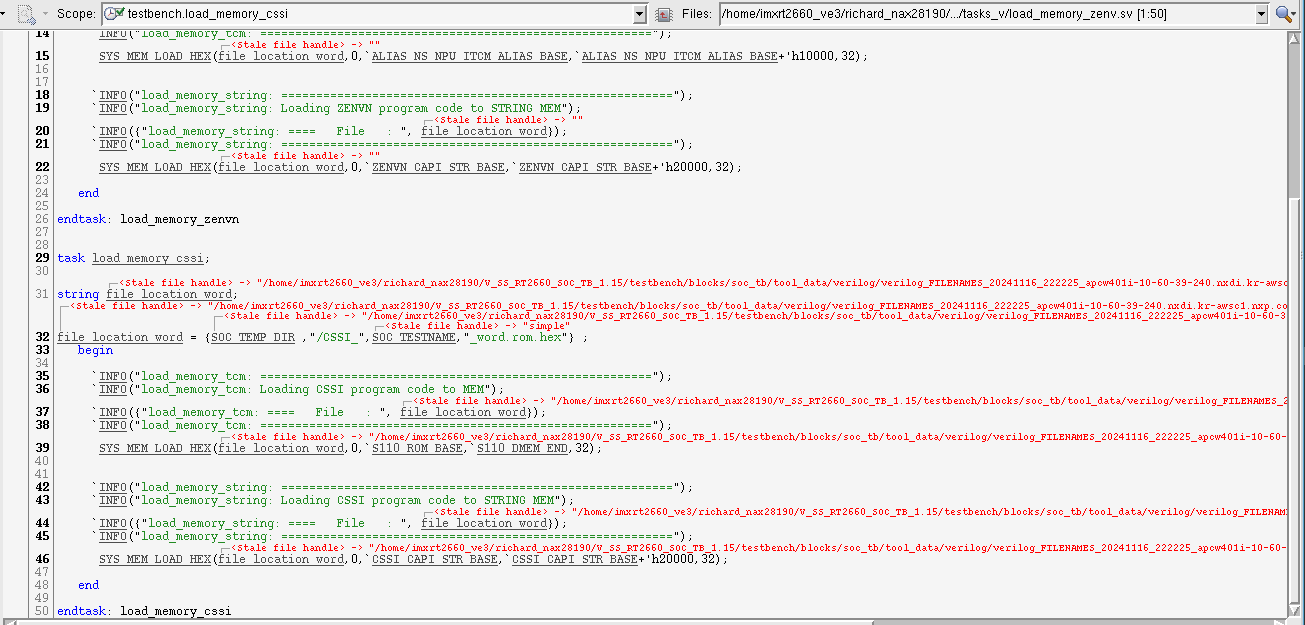
Task\_v/tb\_mem\_tasks.sv



### Load\_memory\_cssi

Task\_v/load\_memory\_zenv.sv

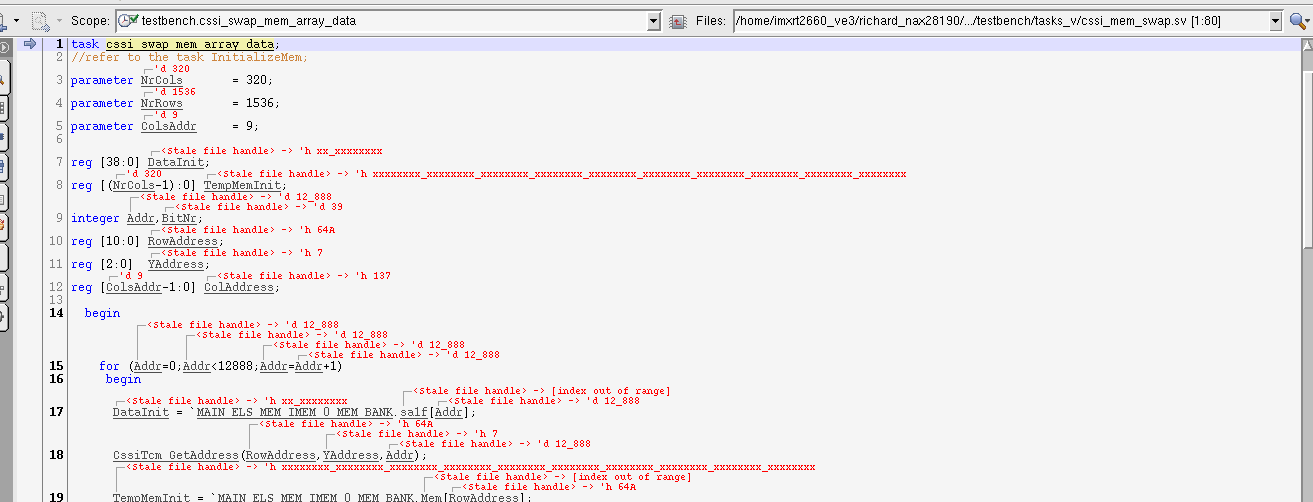
Load hex文件到CSSI memory中



### Cssi\_swap\_mem\_array\_data

Tasks\_v/cssi\_mem\_swap.sv

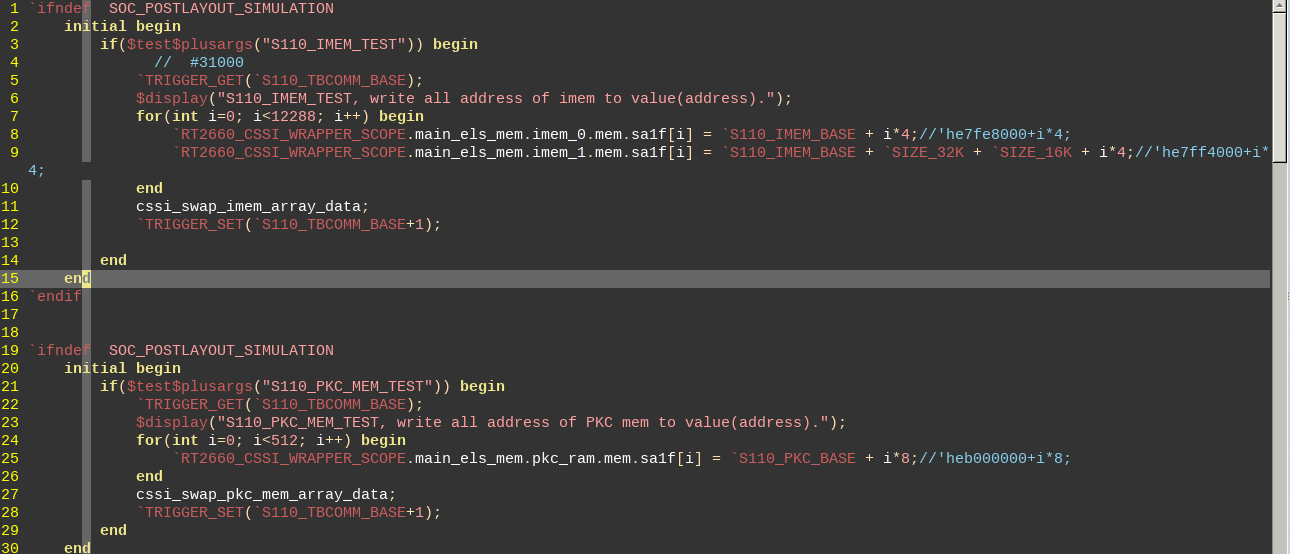
将RAM sa1f中的数据swap到RAM MEM中。Cpu从RAM MEM中获取数据



### S110\_RAM\_INIT

PRELOAD RAM data for testing mapping

testbench/blocks/soc\_tb/testbench/instances\_v



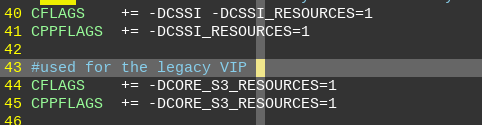
## C架构

### CSSI\_RESOURCES/ DCORE\_S3\_RESOURCES

如下两个宏用来识别属于S110的编译内容

CSSI\_RESOURCES/ DCORE\_S3\_RESOURCES(for legacy VIP code from imx)

testbench/blocks/soc\_tb/tool\_data/compiler/zenv/cssi/makefile.pre\_compile



### Environment.h

Test case C 文件中包含environment.h

testbench/blocks/soc\_tb/tool\_data/compiler/include/environment.h



### cssi\_environment.h

S110的case会选择cssi\_environemnt

testbench/blocks/soc\_tb/tool\_data/compiler/zenv/cssi/include/cssi\_environment.h

#include <context\_cssi\_api.h> // 支持的IP context

#include <pre\_post\_main.h>

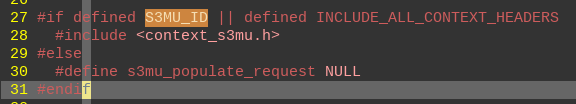
### context\_cssi\_api.h

context 架构， 需要跟cssi\_ip\_request.h配对

当C环境中有S3MU\_ID这个宏时， MU VIP 中的context\_s3mu.h会被编译

一般定义在arg文件中，如果环境需要这个context默认被编译的时候，可以加到environment.h中。

VIP中的头文件被包含进去

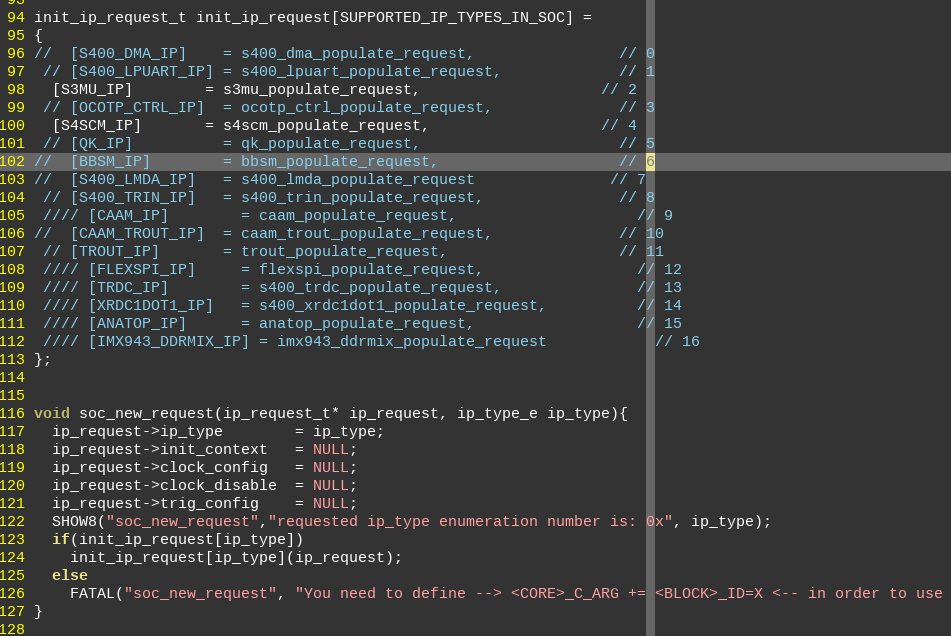


#### Soc\_new\_request

当调用函数soc\_new\_request的时候。

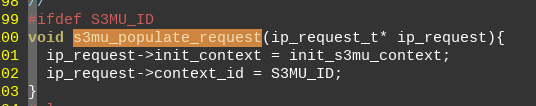
会根据ip\_type 选择对应的function进行populate函数赋值操作。

Ip\_type的定义在s3\_ip\_request中



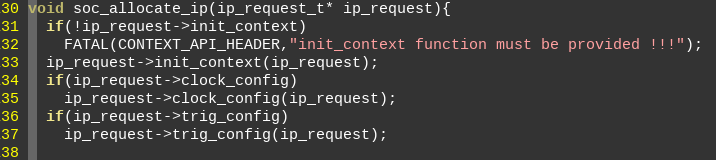
#### 执行populate函数

S3MU VIP中，populate的操作其实也是在进行赋值操作



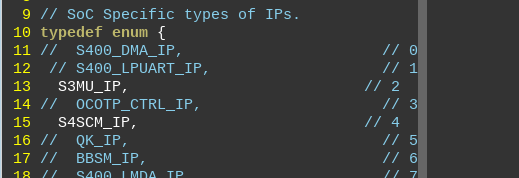
#### Allocate

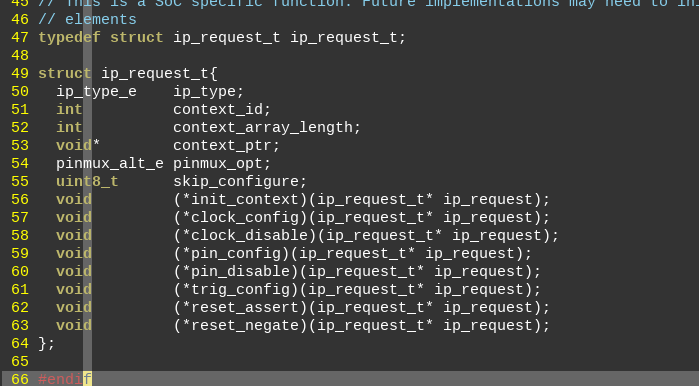
执行allocate的目的是执行如上new\_request获取的函数，

包括create context， clock ， reset， PIN 配置。

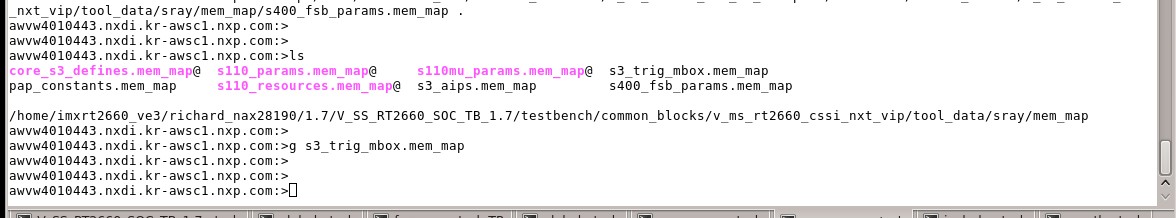
### cssi\_ip\_request.h

定义了相关module的宏，已经ip\_request结构体





## MU



### 定义

Context内宏定义如下



Cssi\_vip/s3mu\_params.mem\_map

S3MU0\_RBASE -> `S3MU0\_A\_RBASE/S3MU0\_B\_RBASE -> S400\_HOST\_ADR\_S3\_MU\_APP\_RBASE/S400\_LOCAL\_PERIPHERALS\_MUB\_RBASE

CSSI\_VIP/s400\_params.mem\_map

S400\_HOST\_ADR\_S3\_MU\_APP\_RBASE (`SENTINEL\_BASE\_ADDRESS + 32'h00020000)

Soc-mem-map/sys\_base\_address.mem\_map:16:

define SENTINEL\_BASE\_ADDRESS 'h4750\_0000

## mem\_map

define all the S110 internal IP base address:

v\_ms\_rt2660\_cssi\_nxt\_vip/tool\_data/sray/mem\_map/s110\_params.mem\_map

### CSSI DMEM

### IMEM

