1. Tìm ra stub function -> local function
2. Tìm ra globlal variable (kiểm tra các giá trị input và output của các hàm nhỏ xem có lấy ở global không)
3. Tìm ra input parameter và return value
4. Với pointer thì range phù hợp là not NULL

//Trong r\_impdrv\_api.c

e\_impdrv\_errorcode\_t R\_IMPDRV\_SetPMPolicy ()

có hàm

impdrv\_api\_chk\_core\_info

impdrv\_api\_cnv\_ctrl\_handle

impdrv\_genctl\_pm\_set\_policy //hàm xử lý chính trong r\_impdrv\_genctl.c



impdrv\_api\_cnv\_error\_code

3 hàm kia chỉ đơn giản nhận input từ hàm **R\_IMPDRV\_SetPMPolicy** và trả về error code, ta tập trung vào hàm

**impdrv\_genctl\_pm\_set\_policy**:

Trong đây có :

impdrv\_genctl\_prologue

impdrv\_genctl\_chk\_core\_info

impdrv\_cmnctl\_mutex\_lock  
 impdrv\_osdep\_mutex\_lock  
 impdrv\_cmnctl\_test\_checksum  
 impdrv\_cmnctl\_reg\_test\_checksum  
 impdrv\_osdep\_mutex\_unlock

trong đây sử dụng hàm có truyền vào biến global variable **g\_impdrv\_cmn\_ctl\_ins**, đây là struct chứa **mutex\_handle** và **mutex\_time\_period**

gọi hàm **R\_OSAL\_ThsyncMutexLockForTimePeriod** , đây là stub function, sẽ đưa vào trong file excel

impdrv\_genctl\_epilogue

impdrv\_cmnctl\_mutex\_unlock

impdrv\_osdep\_mutex\_unlock

gọi hàm **R\_OSAL\_ThsyncMutexUnlock**, đây là stub function, sẽ đưa vào trong file excel

impdrv\_cmnctl\_get\_corectl\_func (p\_core\_info->core\_type)

trong đây có định nghĩa 1 func pointer

typedef struct

{

    bool (\*p\_impdrvCorectlIsValidCore)(

        const uint32\_t  core\_num

    );  /\*\*< Function table for distinguishing each core. Valid core.   \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlCheckState)(

        const st\_impdrv\_corectl\_t   \*const p\_corectl,

        const uint32\_t              core\_num,

        const e\_impdrv\_state\_t      state

    );  /\*\*< Function table for distinguishing each core. Check state.  \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetState)(

        st\_impdrv\_corectl\_t         \*const p\_corectl,

        const e\_impdrv\_state\_t      state

    );  /\*\*< Function table for distinguishing each core. Set state.    \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlInitStart)(

        st\_impdrv\_coretypectl\_t     \*const p\_coretypectl,

        const uint32\_t              core\_num

    );  /\*\*< Function table for distinguishing each core. Init start.   \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlInitEnd)(

        st\_impdrv\_coretypectl\_t     \*const p\_coretypectl,

        const uint32\_t              core\_num

    );  /\*\*< Function table for distinguishing each core. Init end.     \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlStart)(

        st\_impdrv\_coretypectl\_t     \*const p\_coretypectl,

        const uint32\_t              core\_num

    );  /\*\*< Function table for distinguishing each core. Start core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlStop)(

        st\_impdrv\_coretypectl\_t     \*const p\_coretypectl,

        const uint32\_t              core\_num

    );  /\*\*< Function table for distinguishing each core. Stop core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlAttInit)(

        st\_impdrv\_corectl\_t         \*const p\_corectl,

        const uint32\_t              core\_num

    );  /\*\*< Function table for distinguishing each core. Attribute initialize. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetMemInit)(

        st\_impdrv\_corectl\_t     \*const p\_corectl,

        const uint32\_t          core\_num,

        const e\_impdrv\_param\_t  enable

    );  /\*\*< Function table for distinguishing each core. Set core memory initialize.   \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetCoreMap)(

        st\_impdrv\_corectl\_t     \*const p\_corectl,

        const uint32\_t          core\_num,

        const uint8\_t           syncc\_val[IMPDRV\_COREMAP\_MAXID]

    );  /\*\*< Function table for distinguishing each core. Set sync core map.    \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetCl)(

        st\_impdrv\_corectl\_t     \*const p\_corectl,

        const uint32\_t          core\_num,

        const uint32\_t          claddr\_phys

    );  /\*\*< Function table for distinguishing each core. Set CL address.   \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetIrqMask)(

        st\_impdrv\_corectl\_t     \*const p\_corectl,

        const uint32\_t          core\_num,

        const bool              irq\_mask[IMPDRV\_IRQMASK\_MAX]

    );  /\*\*< Function table for distinguishing each core. Set IRQ mask setting. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlExecute)(

        st\_impdrv\_corectl\_t     \*const p\_corectl,

        const uint32\_t          core\_num,

        const p\_impdrv\_cbfunc\_t callback\_func,

        void                    \*const p\_callback\_args

    );  /\*\*< Function table for distinguishing each core. Execute core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlIntHandler)(

        st\_impdrv\_corectl\_t     \*const p\_corectl,

        const uint32\_t          core\_num

    );  /\*\*< Function table for distinguishing each core. int handler imp control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlResumeExe)(

        st\_impdrv\_corectl\_t     \*const p\_corectl,

        const uint32\_t          core\_num

    );  /\*\*< Function table for distinguishing each core. Resume core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlPmSetPolicy)(

        st\_impdrv\_coretypectl\_t        \*const p\_coretypectl,

        const uint32\_t                  core\_num,

        const e\_impdrv\_pm\_policy\_t      policy

    );  /\*\*< Function table for distinguishing each core. Pm set policy core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlPmGetPolicy)(

        st\_impdrv\_coretypectl\_t         \*const p\_coretypectl,

        const uint32\_t                  core\_num,

        e\_impdrv\_pm\_policy\_t            \*const p\_policy

    );  /\*\*< Function table for distinguishing each core. Pm get policy core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetDsp)(

        st\_impdrv\_corectl\_t         \*p\_corectl,

        const uint32\_t              core\_num,

        const st\_impdrv\_dsp\_data\_t  \*const p\_dsp\_app,

        const st\_impdrv\_dsp\_data\_t  \*const p\_dsp\_fw,

        const st\_impdrv\_dsp\_data\_t  \*const p\_dsp\_data,

        const st\_impdrv\_dsp\_data\_t  \*const p\_dsp\_dtcm

    );  /\*\*< Function table for distinguishing each core. Pm get policy core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlBusIfCheck)(

        st\_impdrv\_coretypectl\_t         \*const p\_coretypectl,

        const uint32\_t                  core\_num,

        const st\_impdrv\_chk\_resource\_t  \*const p\_chk\_resource

    );  /\*\*< Function table for distinguishing each core. Interface bus check of core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlConfRegCheck)(

        const st\_impdrv\_coretypectl\_t   \*const p\_coretypectl,

        const uint32\_t                  core\_num,

        const st\_impdrv\_chk\_param\_t     \*const p\_chk\_param,

        const uint32\_t                  param\_num

    );  /\*\*< Function table for distinguishing each core. Configuration register check of core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetClBrkAddr)(

        st\_impdrv\_corectl\_t     \*const p\_corectl,

        const uint32\_t          core\_num,

        const uint32\_t          cl\_brk\_addr

    );  /\*\*< Function table for distinguishing each core. Set CL break address.   \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSetGosubCond)(

        st\_impdrv\_corectl\_t         \*const p\_corectl,

        const uint32\_t              core\_num,

        const e\_impdrv\_gosub\_cond\_t condition

    );  /\*\*< Function table for distinguishing each core. Set Conditional GOSUB instruction attribute.  \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlPmGetState)(

        st\_impdrv\_coretypectl\_t     \*const p\_coretypectl,

        const uint32\_t              core\_num,

        e\_impdrv\_pm\_state\_t         \*const p\_pmstate

    );  /\*\*< Function table for distinguishing each core. Pm get state core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlQuit)(

        const uint32\_t              core\_num

    );  /\*\*< Function table for distinguishing each core. Quit core control. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSyncStart)(

        st\_impdrv\_coretypectl\_t coretype\_ctl[IMPDRV\_CORE\_TYPE\_MAX],

        const uint32\_t          core\_num

    );  /\*\*< Function table for distinguishing each core. Starts controlling the core in sync. \*/

    e\_impdrv\_errorcode\_t (\*p\_impdrvCorectlSyncStop)(

        st\_impdrv\_coretypectl\_t coretype\_ctl[IMPDRV\_CORE\_TYPE\_MAX],

        const uint32\_t          core\_num

    );  /\*\*< Function table for distinguishing each core. Stop controlling the core in sync. \*/

} st\_impdrv\_corectl\_func\_t;

Ở đây gồm 26 function pointer (chứa địa chỉ của func) trong struct này

Bên cạnh đó hàm này thực thi truyền vào 26 function pointer vào trong struc **func,** ví dụ khi truyền core\_type là IMP thì sẽ nhận được 26 function sau vào struct **func**

/\*\* IMPDRV\_CORE\_TYPE\_IMP \*/

{   impdrv\_impctl\_is\_valid\_core,    impdrv\_impctl\_check\_state,      impdrv\_impctl\_set\_state,

    impdrv\_impctl\_init\_start,       impdrv\_impctl\_init\_end,         impdrv\_impctl\_start,

    impdrv\_impctl\_stop,             impdrv\_impctl\_attr\_init,        impdrv\_impctl\_set\_mem\_init,

    impdrv\_impctl\_set\_core\_map,     impdrv\_impctl\_set\_cl,           impdrv\_impctl\_set\_irq\_mask,

    impdrv\_impctl\_execute,          impdrv\_impctl\_int\_handler,      impdrv\_impctl\_resume\_exe,

impdrv\_impctl\_pm\_set\_policy,    impdrv\_impctl\_pm\_get\_policy,    NULL,

    NULL,                           NULL,                           impdrv\_impctl\_set\_cl\_brk\_addr,

    impdrv\_impctl\_set\_cond\_gosub,   impdrv\_impctl\_pm\_get\_state,     impdrv\_impctl\_quit,

    NULL,                           NULL

},

Sau khi sao chép xong thì hàm sẽ lấy đúng vị trí **p\_impdrvCorectlPmSetPolicy** để thực thi (như ta thấy trong struct thì hàm này nằm ở vị trí 15 tính từ 0)

ercd = p\_funcs->p\_impdrvCorectlPmSetPolicy(…)

tức nó đang thực thi

impdrv\_impctl\_pm\_set\_policy

bao gồm:

impdrv\_cnnctl\_is\_valid\_core //so sánh số core rồi return

impdrv\_osdep\_pm\_set\_policy(trong này xử lý 5 hàm)

impdrv\_osdep\_chk\_core\_info // check core type rồi return

impdrv\_osdep\_dev\_open\_dsp //core\_type = DSP thì vào đây

impdrv\_osdep\_chk\_core\_info // check core type rồi return

impdrv\_osdep\_dev\_open\_dsp\_sub //ko lỗi -> mở core

R\_OSAL\_IoDeviceOpen // gọi stub -> đưa vào excel



impdrv\_osdep\_dev\_open\_imp //core type = những loại còn lại vào đây (IMP)

impdrv\_osdep\_chk\_core\_info // check core type rồi return

impdrv\_osdep\_dev\_open //ko lỗi -> mở core

R\_OSAL\_IoDeviceOpen // gọi stub -> đưa vào excel



R\_OSAL\_PmSetPolicy // gọi stub -> đưa vào excel

A computer screen shot of a policy

Description automatically generated

impdrv\_osdep\_dev\_close // gọi stub -> đưa vào excel

