Implementation of linux SH2

SIOS Technology, Inc.

Yoshinori Sato

<ysato@users.sourceforge.jp>

Contents

- Current status
- A policy
- Implementation
  - Exception handling
  - Peripheral support
- Future working

Current status

- It was done merge with linux-2.6.20-rc1
- Latest version is git repository of linux-sh
- The part which required prepared it as SH2 architecture correspondence minimally
- However, support of individual CPU / target is a future problem

A policy

- It works at once.
- I do merge in linux-sh.
- I use code for existing sh3/4 usefully.
- I dismiss code of old sh2 support.

Implementation

- A difference of SH2(A) and SH3/4
  - Exception handling
  - Handling
  - exception factor
  - Privileged mode
  - Register bank
  - MMU
  - Address space (only SH2A)
- I become a problem when I let linux kernel work on SH2(A).

Exception handling of SH3/4

1. Save Registers (PC,SR,SP)
2. Enter privileged mode
3. SR update
4. Jump to exception handler

Exception handling of SH2

1. Load exception handler address from vector table
2. Save SR to stack
3. Save PC to stack
4. Jump to exception handler

Procedure to be completely different from SH3/4

SIOS

Exception handling of SH2

- Exception handling of SH2 is different from SH3/4.
- That purpose cannot make a common use of SH3/4 with exception handling.
- And cannot support request of generic interrupt handler.

Exception handling of SH2

- This problem is settled by converting SH3/4 into with an exception entry in the same way.
- It is substantially forcible implementation.

Exception handling of SH2

- Convert an vector address to number
- Execute the following code with all exception vector.

```
exception_entry:
            mov.l       r0,@-sp
            mov #no,r0 ! <- set number here
            bra exception_trampoline
            and #0xff,r0
```

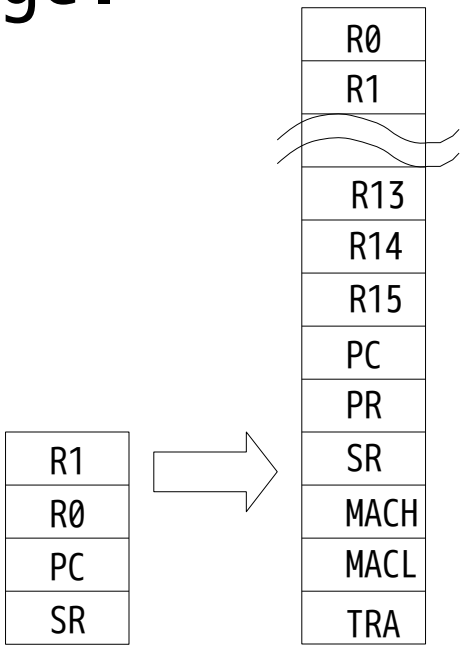
SIOS

Exception handling of SH2

- Can set exception number in r0 therefore.
- I have only wordy implementation on by a restriction of instruction set.

Exception handling of SH2

- Generation of exception stack frame
- I convert it into the stack frame which is compatible with SH3/4 with this stage.



SIOS

Exception handling of SH2

- A factor of an exception
- Identification number is assigned to SH3/4 by an exception / trap / interrupt independently.
- (EXPEVT/TRA/INTEVT)
- However, I do not become independent in SH2.

Exception handling of SH2

- I classed a factor and did an assign as follows

0x00 - 0x1f exception

0x20 - 0x3f trap (system call)

0x40 - 0xff interrupt

System call

- I assigned system call from 0x20 to 0x2f of trap in SH2.
- However, use 0x10 to 0x1f in SH3/4.
- I examined that I assigned the same number.
- Because there were a few advantages by supporting, I gave priority to easiness of implementation.

SIOS

System call

- I lose binary compatible nature by this.
- Because I cannot work an usual ELF binary in nommu, this thinks that there is not it in a problem.
- I make it source compatible with SYSCALL macro.

Privileged mode

- There is not such a features in SH2.
- I implement similar features with software with that purpose.
- I use mode judgment flag and do a mode judgment / conversion in exception appearance.

Privileged mode

- I thought about a method to judge in address of PC.
- Because a judgment of kernel module became complicated, I did not adopt it.

Privileged mode

```
ENTRY(exception_handler)
        mov.l   r2,@-sp
        mov.l   r3,@-sp
        mov     r0,r1
        cli
        mov.l   $cpu_mode,r2    <- Privilege / User Flag
        mov.l   @r2,r0
        mov.l   @(5*4,r15),r3   <- original SR
        shll2   r3
        rotl    r0
        rotl    r0              <- mode flag to T
        rotcr   r3
        shlr    r3  <- Privilege / User Flag to original SR
        shlr    r0
        bt/s    1f
         mov.l  r3,@(5*4,r15)
```

SIOS

Register bank

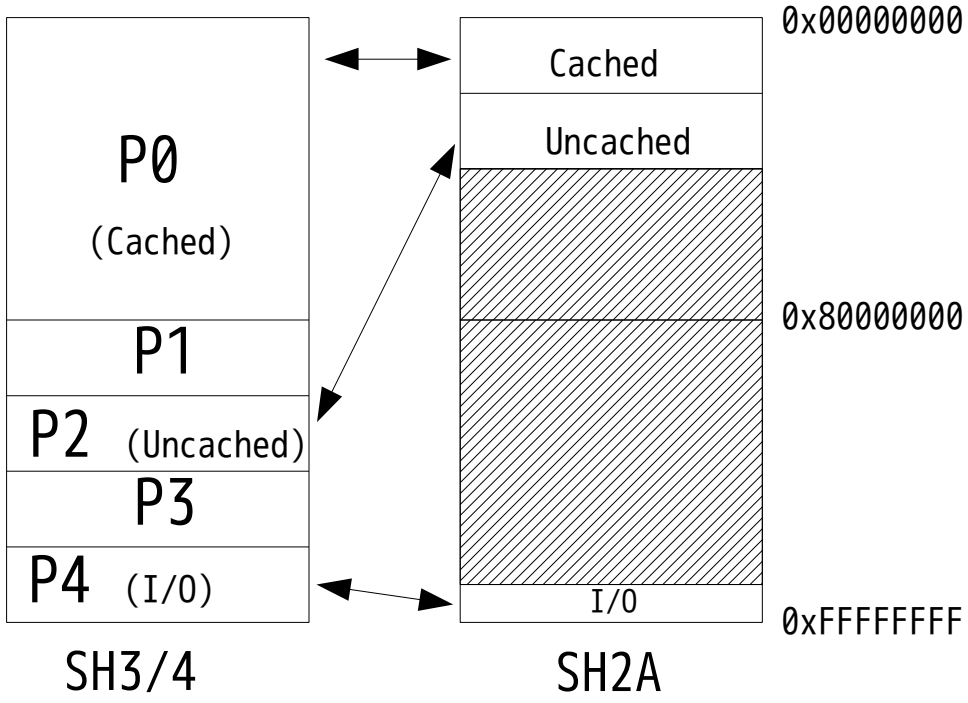
- As for SH3/4, R0-R7 becomes bank structure for privileged mode.
- Bank register is not prepared in SH2.
- (The thing that register bank of SH2A is wrong.)
- Because there is config(CONFIG_SR_RB=n) which I do not use bank for in linux-sh, I use it.
- Because not complete, I did some revisions.

MMU

- MMU is not supported in SH2.
- Because the implementation that does not use MMU for linux-sh already is prepared, I use it.
- Because not complete, I did some revisions.

Address space

- SH2A is different from SH3/4 in definitions of address space.
- I let you be equivalent as follows.



SIOS

Peripheral support

I use one driver already.

Because some peripheral was material different from SH3/4

I made those driver anew.

- Interval timer
- Interrupt controler
- ...

SIOS

Future working

- Cleanup and optimization of code
- Stage to work at once completes
- Don't use enhanced function of SH2A
- It supports more targets

Thanks

- Paul Mundt
- Ishiwatari-san
- Munakata-san

SIOS

Any Questions?

Because I am weak in English,

please talk slowly _(mm)_

SIOS