############################

## test\_config.h

############################

## SHORT\_TEST1, NOP\_INSERT, TEST1~9,

############################

## test\_define.h

############################

## CODEBASE, DATABASE, NOP, NOP4, LI, LI\_EXIMM

## TLBREBASE, TLB\_ENTRY, ……

#define CODEBASE 0x10000

#define DATABASE 0x1d0000

#if NOP\_INSERT

#define NOP addi.w zero, zero, 0

#else

#define NOP

#endif

#define NOP4 NOP;NOP;NOP;NOP

#define LI (reg, imm) \

lu12i.w reg , (((imm>>12) + ((imm&0x00000800)>>11)) & 0x000fffff) & 0x80000? (((imm>>12) + ((imm&0x00000800)>>11)) & 0x000fffff) - 0x100000: (((imm>>12) + ((imm&0x00000800)>>11)) & 0x000fffff); \

NOP4; \

addi.w reg, reg, (imm & 0x00000fff)&0x800?(imm & 0x00000fff)-0x1000:(imm & 0x00000fff)

#define LI\_EXIMM (reg1,reg2,imm) \

li.w reg1, DATABASE; \

li.w reg2, imm; \

st.w reg2, reg1, 0

############################

## cpu\_cde.h:

############################

## LED\_ADDR, NUM\_ADDR, ……

#define TIMER\_ADDR 0xbfafe000

#define LED\_ADDR 0xbfaff020

#define LED\_RG0\_ADDR 0xbfaff030

#define LED\_RG1\_ADDR 0xbfaff040

#define NUM\_ADDR 0xbfaff050

#define SWITCH\_ADDR 0xbfaff060

#define BTN\_KEY\_ADDR 0xbfaff070

#define BTN\_STEP\_ADDR 0xbfaff080

#define SW\_INTER\_ADDR 0xbfaff090 //switch interleave

#define UART\_ADDR 0xbfafff10

############################

## start.S:

############################

## s0, number

## s1, number adress ## NUM\_ADDR

## s2, exception use

## s3, score

## s4, exception pc

.globl start

start:

li.w t0, 0xffffffff

addi.w t0, zero, -1

#ifdef CSR\_INIT # TEST8

ori $r1,$r0,0x28

csrwr $r1,0x0

#endif

b locate

NOP4

.org CODEBASE

locate:

LI (a0, LED\_RG1\_ADDR) ## test\_define.h, cpu\_cde.h

LI (a1, LED\_RG0\_ADDR)

LI (a2, LED\_ADDR)

LI (s1, NUM\_ADDR)

LI (t1, 0x0002)

LI (t2, 0x0001)

LI (t3, 0x0000ffff)

lu12i.w s3, 0

NOP4 ## test\_define.h

st.w t1, a0, 0

st.w t2, a1, 0

st.w t3, a2, 0

st.w s3, s1, 0

lu12i.w s0, 0

NOP4

############################

### TEST1 test

#if TEST1

bl n1\_lu12i\_w\_test # test lu12i.w: n1\_lu12i\_w.S

NOP4

bl idle\_1s

NOP4

bl n2\_add\_w\_test # test add.w: n2\_add\_w.S

NOP4

bl idle\_1s

NOP4

… …

test\_end:

LI (s0, TEST\_NUM)

NOP4

beq s0, s3, 1f

NOP4

LI (a0, LED\_ADDR)

LI (a1, LED\_RG1\_ADDR)

LI (a2, LED\_RG0\_ADDR)

LI (t1, 0x0002)

NOP4

st.w zero, a0, 0

st.w t1, a1, 0

st.w t1, a2, 0

li.w a0, 0x1

b 2f

NOP4

1:

LI (t1, 0x0001)

LI (a0, LED\_RG1\_ADDR)

LI (a1, LED\_RG0\_ADDR)

NOP4

st.w t1, a0, 0

st.w t1, a1, 0

li.w a0, 0

2:

//LI (t1, 0xff)

//LI (t0, UART\_ADDR)

//sw t1, 0(t0)

bl test\_finish

NOP4

.org 0x100

test\_finish:

addi.w t0, t0, 1

LI (t2, UART\_ADDR)

st.w zero, t2, 0

1:

b 1b

NOP4

############################

## n2\_sdd\_w.S

############################

LEAF(n2\_add\_w\_test) ## asm.h

addi.w s0, s0 ,1 ## regdef.h

addi.w s2, zero, 0x0

###test inst

TEST\_ADD\_W(0x1adef300, 0x51dd58de, 0x6cbc4bde) ## ins\_test.h

TEST\_ADD\_W(0x9674ea60, 0xab475792, 0x41bc41f2)

……

###detect exception

bne s2, zero, inst\_error

NOP4

###score ++

addi.w s3, s3, 1

inst\_error:

###output (s0<<24)|s3

slli.w t1, s0, 24

NOP4

or t0, t1, s3

NOP4

st.w t0, s1, 0 ## display number

jirl zero, ra, 0

NOP4

END(n2\_add\_w\_test)

############################

## asm.h:

############################

## LEAF, END

/\* Declare leaf routine. \*/

#define LEAF(symbol) \

.text; \

.globl symbol; \

.align 3; \

cfi\_startproc ; \

.type symbol, @function; \

symbol:

#define END(function) \

cfi\_endproc ; \

.size function, .-function;

############################

## ins\_test.h:

############################

## TEST\_LU12I\_W, TEST\_ADD\_W, … …

/\* 1 \*/

#define TEST\_LU12I\_W(in\_a, ref\_base) \

lu12i.w a0, ref\_base&0x80000?ref\_base-0x100000:ref\_base; \

lu12i.w t0, in\_a&0x80000?in\_a-0x100000:in\_a; \

NOP4; \

add.w a0, a0, t1; \

add.w t1, t1, t2; \

NOP4; \

bne t0, a0, inst\_error; \

NOP4

/\* 2 \*/

#define TEST\_ADD\_W(in\_a, in\_b, ref) \

LI (t0, in\_a); \

LI (t1, in\_b); \

LI (v1, ref); \

NOP4; \

add.w v0, t0, t1; \

NOP4; \

bne v0, v1, inst\_error; \

NOP4

############################

## regdef.h:

############################

## zero, a0-7, t0-8, s0-8, ra, sp, ……

## csr\_crmd, csr\_prmd, ……

# define zero $r0

# define ra $r1

# define tp $r2

# define sp $r3

# define a0 $r4

# define a1 $r5

# define a2 $r6

# define a3 $r7

# define a4 $r8

# define a5 $r9

# define a6 $r10

# define a7 $r11

# define v0 $r10 // $r4

# define v1 $r11 // $r5

# define t0 $r12

# define t1 $r13

# define t2 $r14

# define t3 $r15

# define t4 $r16

# define t5 $r17

# define t6 $r18

# define t7 $r19

# define t8 $r20

# define x $r21

# define fp $r22

# define s0 $r23

# define s1 $r24

# define s2 $r25

# define s3 $r26

# define s4 $r27

# define s5 $r28

# define s6 $r29

# define s7 $r30

# define s8 $r31