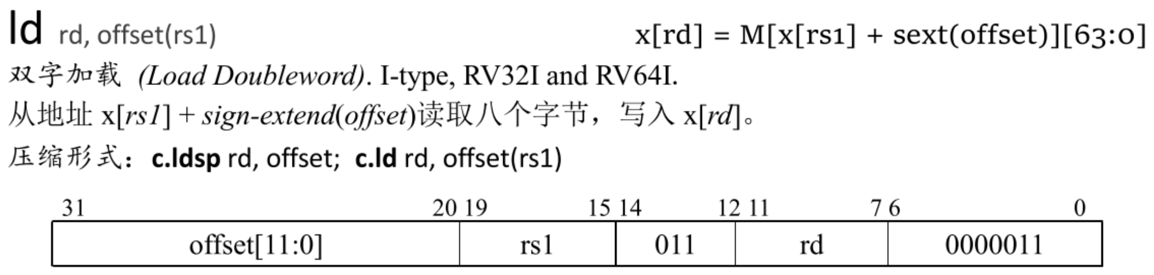
问题报告：

程序现在在中途执行失败，失败原因在于我们编译出来的ld指令与ux607系统自带的ld指令格式不同：

RISC-V指令集中ld指令的格式：



ld rd, offset(rs1)　　　　x[rd] = M[x[rs1] + sext(offset)][63:0]

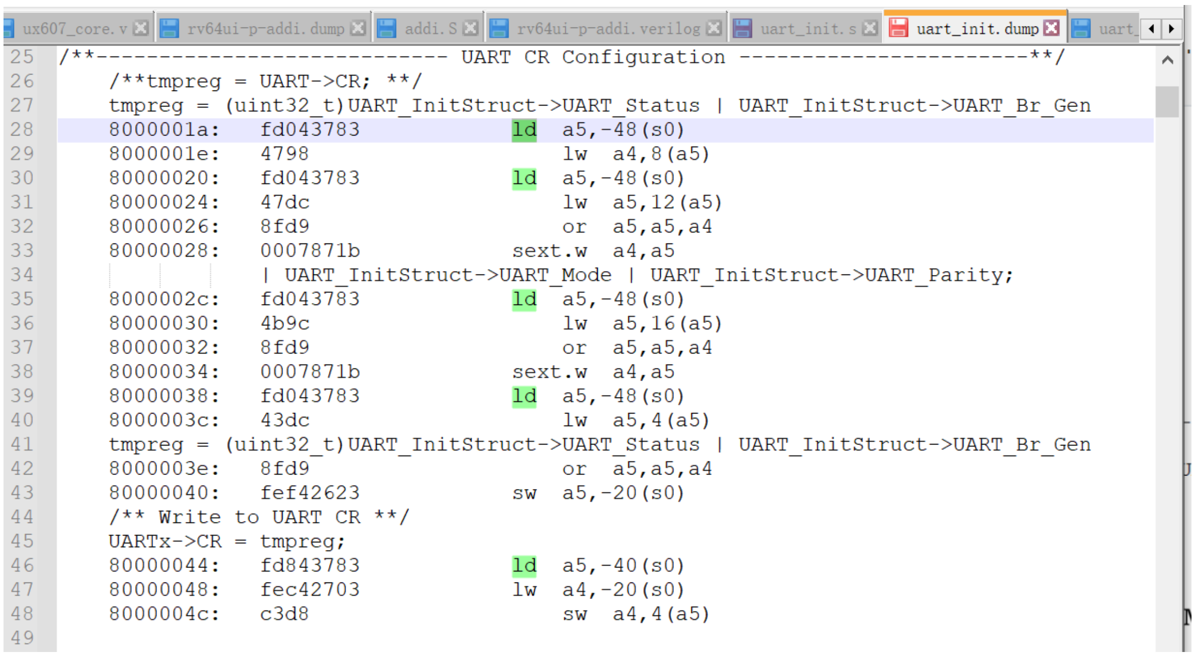
双字加载 *(Load Doubleword)*. I-type, RV32I and RV64I.

从地址 x[*rs1*] + *sign-extend*(*offset*)读取八个字节，写入 x[*rd*]。

压缩形式：**c.ldsp** rd, offset; **c.ld** rd, offset(rs1)

31　　　　　　　　　20	19　　　15	14　　12	11　　　7	6　　　　　0
offset[11:0]	rs1	011	rd	0000011

我们编译出来的ld指令与RISC-V指令集格式完全一致：



```
/**---------------------------- UART CR Configuration -----------------------**/
    /**tmpreg = UART->CR; **/
    tmpreg = (uint32_t)UART_InitStruct->UART_Status | UART_InitStruct->UART_Br_Gen
    8000001a:    fd043783            ld  a5,-48(s0)
    8000001e:    4798                    lw  a4,8(a5)
    80000020:    fd043783            ld  a5,-48(s0)
    80000024:    47dc                    lw  a5,12(a5)
    80000026:    8fd9                    or  a5,a5,a4
    80000028:    0007871b            sext.w  a4,a5
                 | UART_InitStruct->UART_Mode | UART_InitStruct->UART_Parity;
    8000002c:    fd043783            ld  a5,-48(s0)
    80000030:    4b9c                    lw  a5,16(a5)
    80000032:    8fd9                    or  a5,a5,a4
    80000034:    0007871b            sext.w  a4,a5
    80000038:    fd043783            ld  a5,-48(s0)
    8000003c:    43dc                    lw  a5,4(a5)
    tmpreg = (uint32_t)UART_InitStruct->UART_Status | UART_InitStruct->UART_Br_Gen
    8000003e:    8fd9                    or  a5,a5,a4
    80000040:    fef42623            sw  a5,-20(s0)
    /** Write to UART CR **/
    UARTx->CR = tmpreg;
    80000044:    fd843783            ld  a5,-40(s0)
    80000048:    fec42703            lw  a4,-20(s0)
    8000004c:    c3d8                    sw  a4,4(a5)
```

其中：

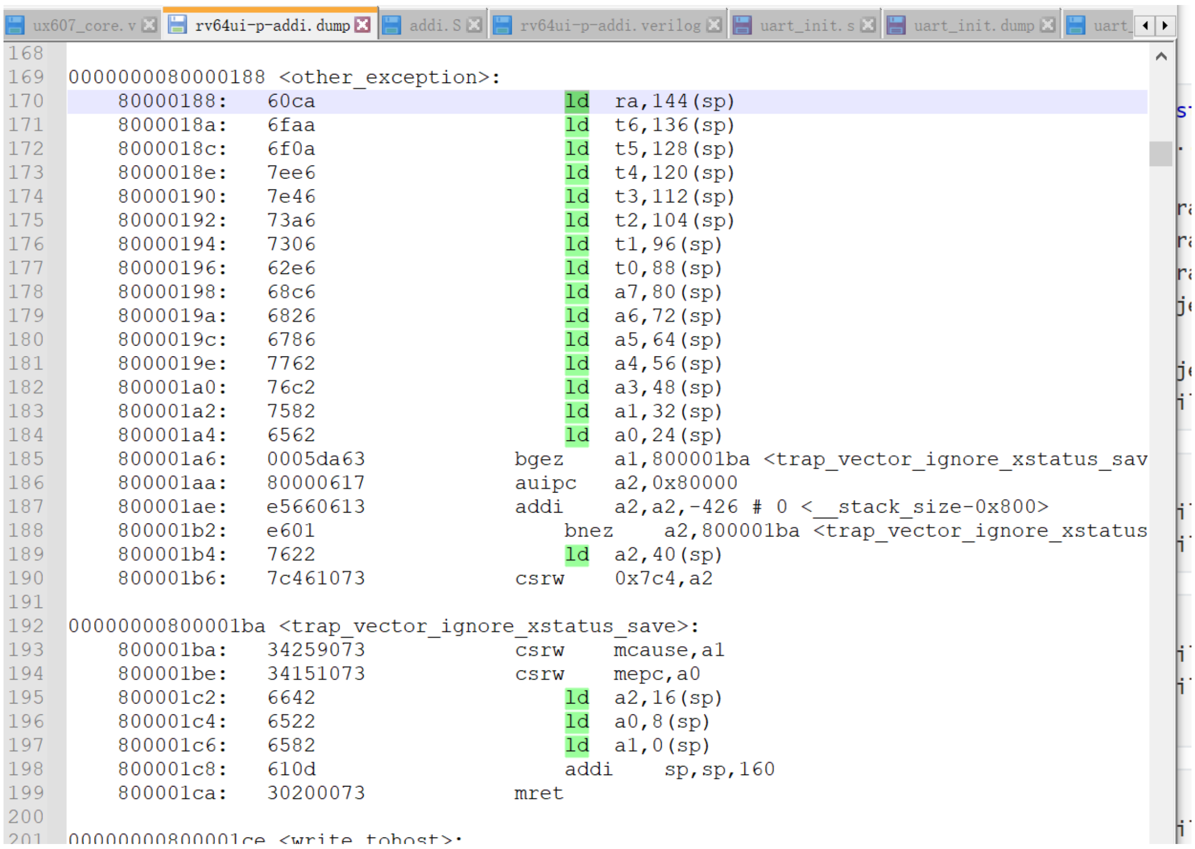
opcode=7'b0000011

rd = 5'b01111;

rs1=5'b01000;

offset=12'hfd0;

但是程序在执行到ld指令后就停止了，然后我查看了ux607自带的testcase中的.dump文件,发现他们的ld格式与我们现有编译出来的ld格式差距很大，如下：



```

0000000080000188 <other_exception>:
    80000188:	60ca                	ld	ra,144(sp)
    8000018a:	6faa                	ld	t6,136(sp)
    8000018c:	6f0a                	ld	t5,128(sp)
    8000018e:	7ee6                	ld	t4,120(sp)
    80000190:	7e46                	ld	t3,112(sp)
    80000192:	73a6                	ld	t2,104(sp)
    80000194:	7306                	ld	t1,96(sp)
    80000196:	62e6                	ld	t0,88(sp)
    80000198:	68c6                	ld	a7,80(sp)
    8000019a:	6826                	ld	a6,72(sp)
    8000019c:	6786                	ld	a5,64(sp)
    8000019e:	7762                	ld	a4,56(sp)
    800001a0:	76c2                	ld	a3,48(sp)
    800001a2:	7582                	ld	a1,32(sp)
    800001a4:	6562                	ld	a0,24(sp)
    800001a6:	0005da63          	bgez	a1,800001ba <trap_vector_ignore_xstatus_sav
    800001aa:	80000617          	auipc	a2,0x80000
    800001ae:	e5660613          	addi	a2,a2,-426 # 0 <__stack_size-0x800>
    800001b2:	e601                	bnez	a2,800001ba <trap_vector_ignore_xstatus
    800001b4:	7622                	ld	a2,40(sp)
    800001b6:	7c461073          	csrw	0x7c4,a2

00000000800001ba <trap_vector_ignore_xstatus_save>:
    800001ba:	34259073          	csrw	mcause,a1
    800001be:	34151073          	csrw	mepc,a0
    800001c2:	6642                	ld	a2,16(sp)
    800001c4:	6522                	ld	a0,8(sp)
    800001c6:	6582                	ld	a1,0(sp)
    800001c8:	610d                	addi	sp,sp,160
    800001ca:	30200073          	mret

00000000800001ce <write_tohost>:
```