**********Draft**********

Debugging hello
test-zipcore & icozip-dev
Using zip-gcc, zip-objdump, verilator & gtkwave
11/04/21

**********Draft**********

The source hello.c was compiled using zip-gcc to create the executeable hello. The program hello.c defiines a structure where a buffer and pointers are stored. Two pointers in bkram 0x0140100c & 0x014010c0 are used to hold pointers to the variables ptrs.w and ptrs.h. The two variable are initialized to 256. With zip-objdump the executeable hello is disasmbled using hello.txt.

When pc-main is used without a <filename> is how it would be used in hardware.

```
./pc-main_tb
Listening on port 8363
Listening on port 8364
> T
        .
        .
        .
Accepted CMD connection
< A04000081Wf
> A04000081K00000000
<  [CLOSED]
Hello, World

  & eewess s


5525hh h6656


BOMB : CPU BREAK RECEIVED
ZIPM--DUMP: Supervisor mode

sR0 : 02000254 sR1 : 0000000d sR2 : 00000000 sR3 : 00c00000
sR4 : 0000000a sR5 : 0200f300 sR6 : 00000000 sR7 : 00000000
sR8 : 00000000 sR9 : 00000000 sR10: 00000000 sR11: 00000000
sR12: 00000000 sSP : 02ffffec sCC : 00000000 sPC : 02000254

uR0 : 00000000 uR1 : 00000000 uR2 : 00000000 uR3 : 00000000
uR4 : 00000000 uR5 : 00000000 uR6 : 00000000 uR7 : 00000000
uR8 : 00000000 uR9 : 00000000 uR10: 00000000 uR11: 00000000
uR12: 00000000 uSP : 00000000 uCC : 00000020 uPC : 02000048

> Z
> Z
```

```
./pc-ziploadd -v ../board/hello
Halting the CPU
Memory regions:
        Block RAM: 01400000 - 01402000
        SDRAM      : 02000000 - 03000000
Loading: ../board/hello
Section 0: 02000000 - 0200f788
Writing to MEM: 02000000-0200f788
Clearing the CPUs registers
Setting PC to 02000000
The CPU should be fully loaded, you may now
start it (from reset/reboot) with:
> wbregs cpu 0

CPU Status is: 00000600
```

Where w is stored

```
./pc-wbregs 0x0140100c
0140100c (        ) : [....] 0200f710

./pc-wbregs 0x0200f710
0200f710 (        ) : [....] 00000100
```

Where h is stored

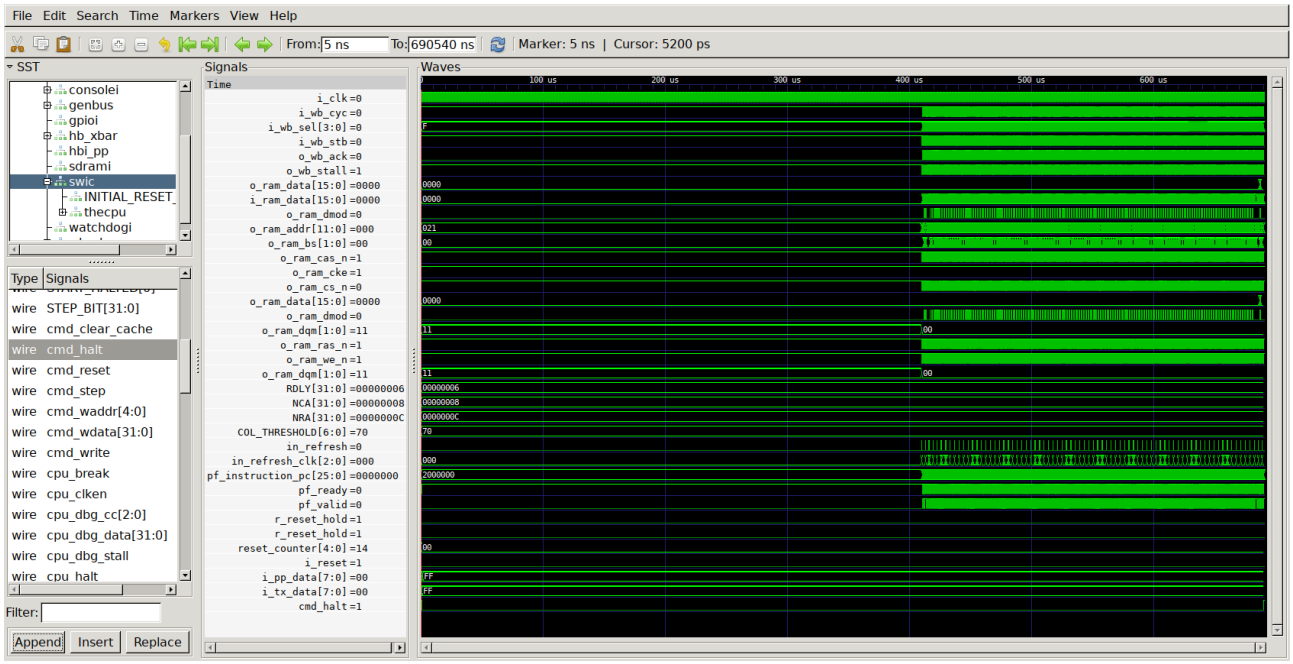
```
./pc-wbregs 0x014010c0
014010c0 (        ) : [....] 0200f714

/pc-wbregs 0x0200f714
0200f714 (        ) : [....] 00000100

./pc-main_tb -d ../../sw/board/hello
```

The overall view of the simulation.

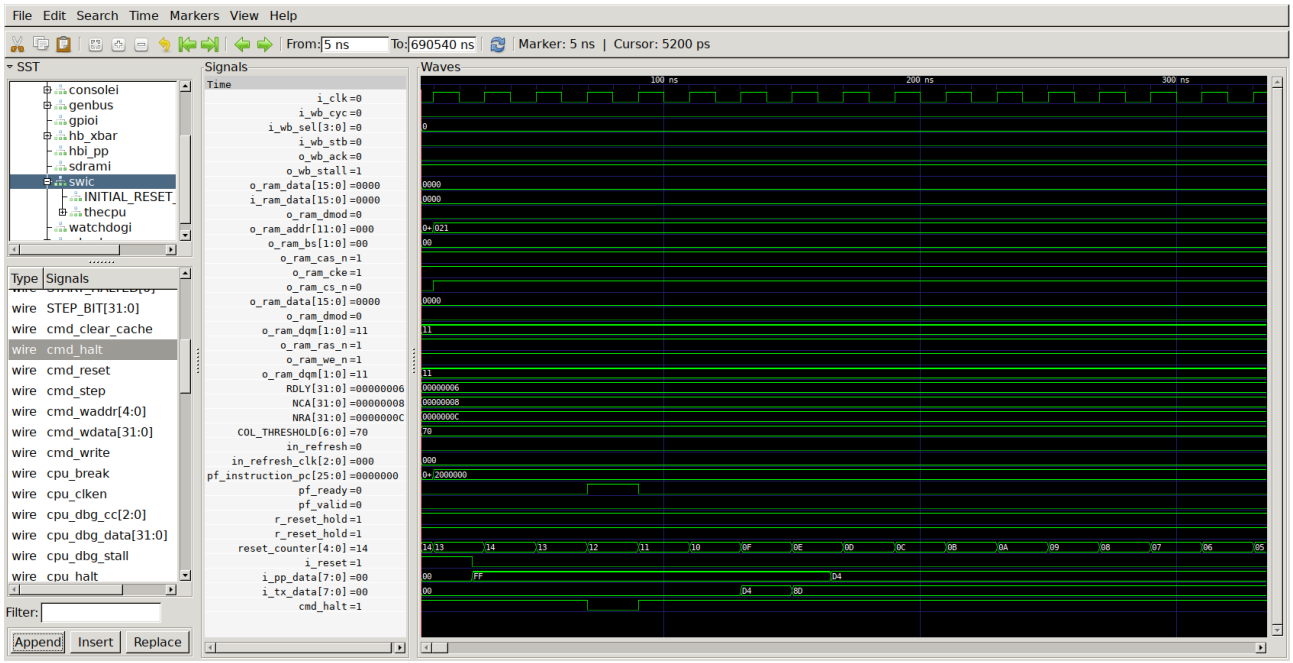
fig1.



With the reset which goes hi then low the address goes from 0x00000000 to 0x0200000.

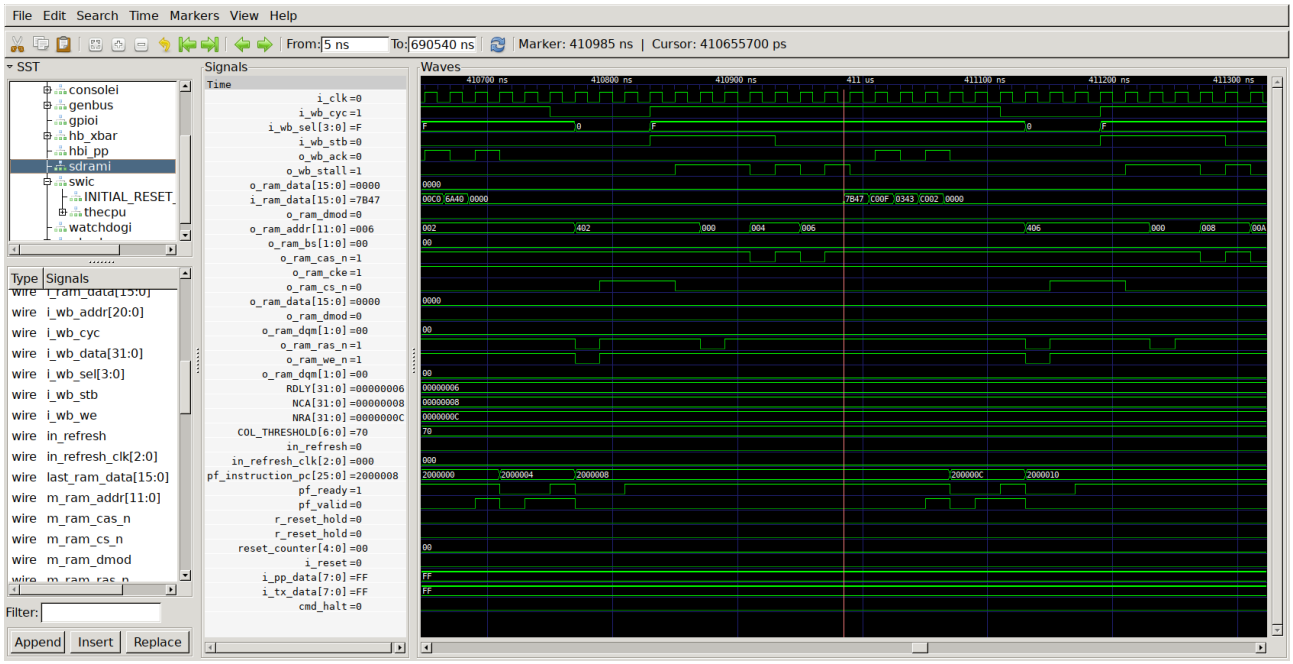
02000000 <_boot_address>:
2000000: 6a 00 00 c0 LDI 0x03000000,SP // 3000000 <_top_of_stack>
2000004: 6a 40 00 00

fig 2.



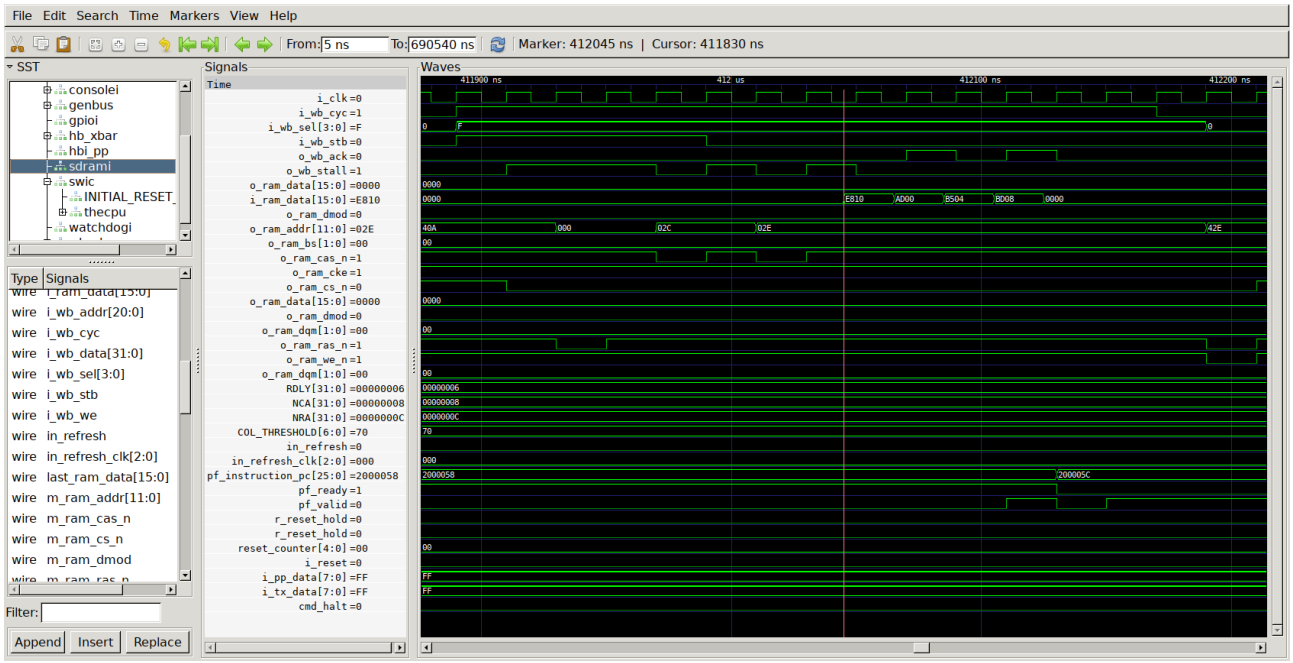
2000008: 7b 47 c0 0f MOV $60+PC,uPC
200000c: 03 43 c0 02 LJSR @0x02000058 // 2000058 <_bootloader>

fig3.



02000058 <_bootloader>:
2000058: e8 10 ad 00 SUB $16,SP | SW R5,(SP)

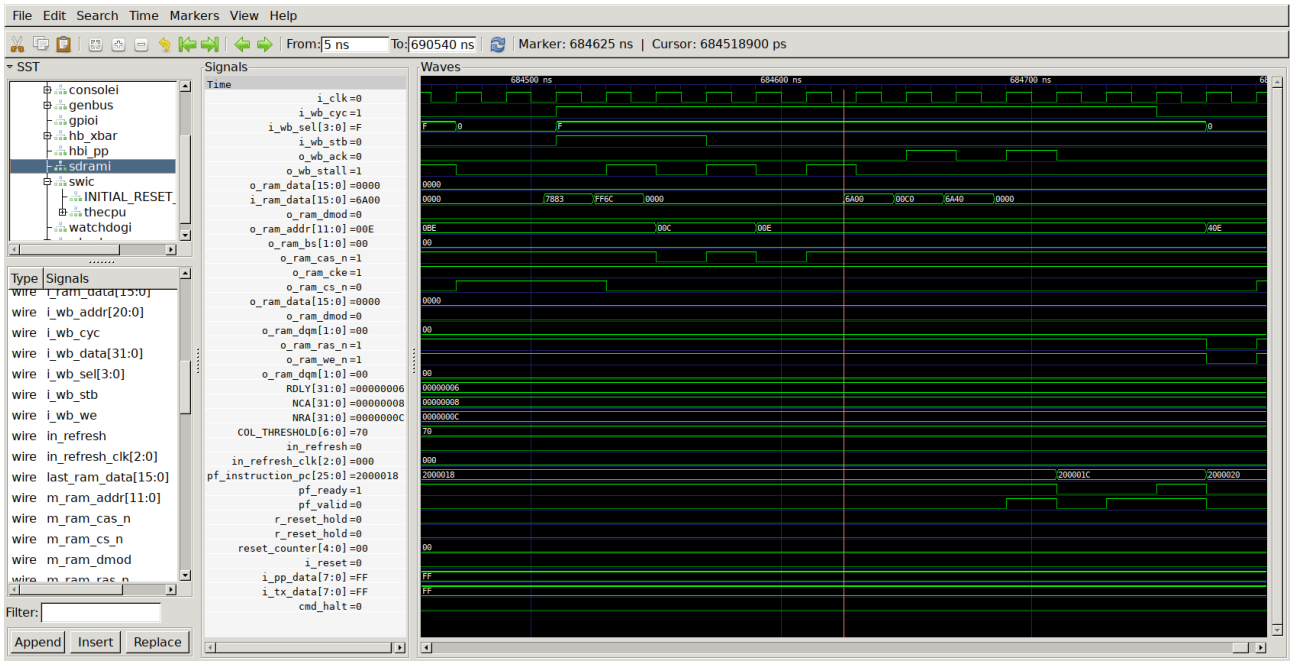
fig4.



Afer the completion of the bootloader.

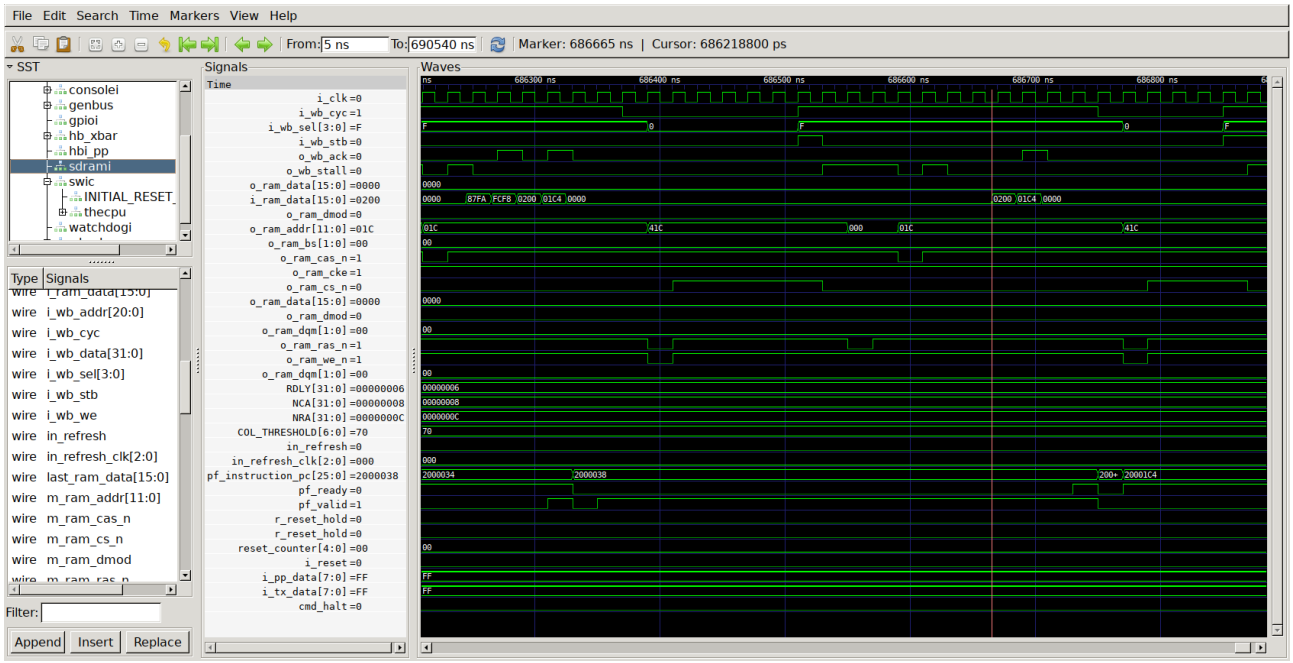
02000018 <_after_bootloader>:
2000018: 6a 00 00 c0 LDI 0x03000000,SP // 3000000 <_top_of_stack>
200001c: 6a 40 00 00

fig5.



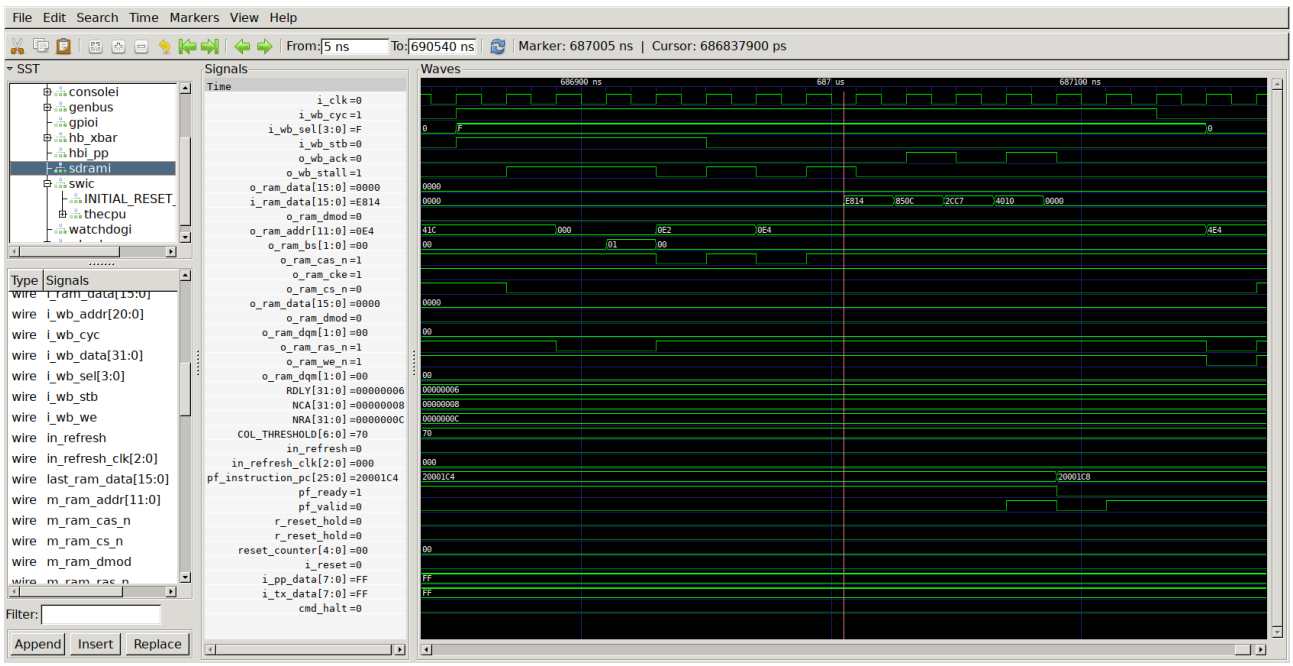
2000034: 87 fa fc f8 JSR 0x020001c4 // 20001c4 <main>
2000038: 02 00 01 c4

fig6.



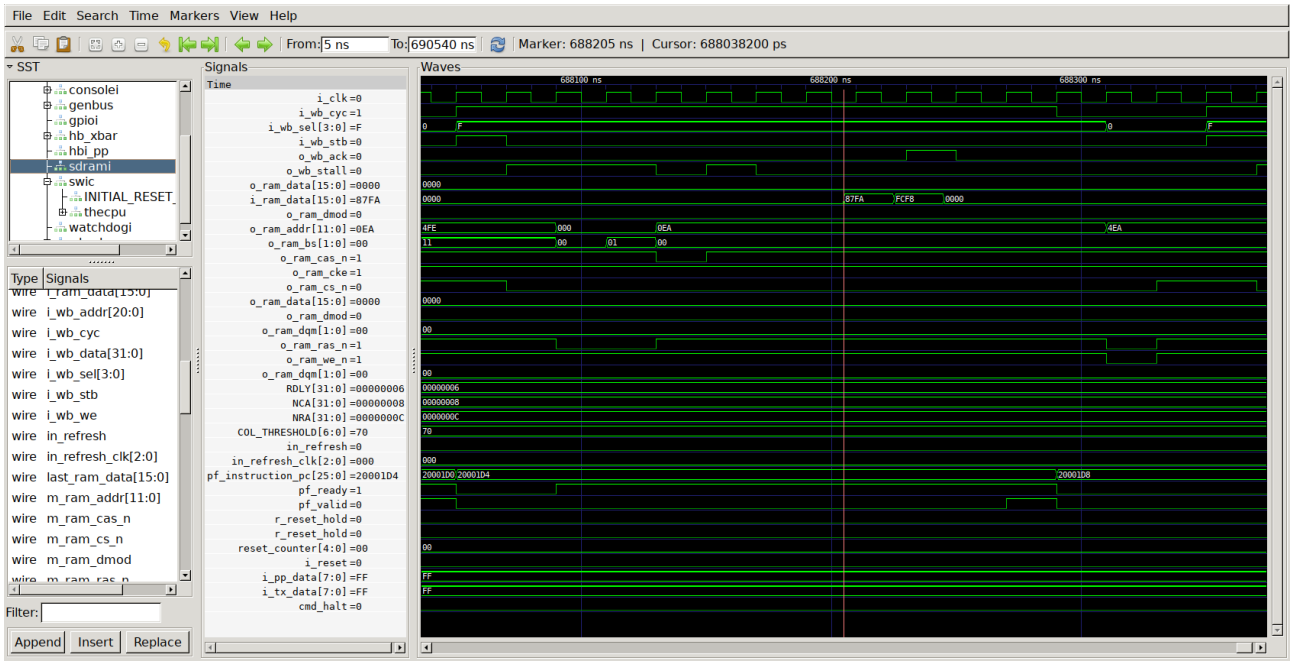
020001c4 <main>:
20001c4: e8 14 85 0c SUB $20,SP | SW R0,$12(SP)
20001c8: 2c c7 40 10 SW R5,$16(SP)

fig7.



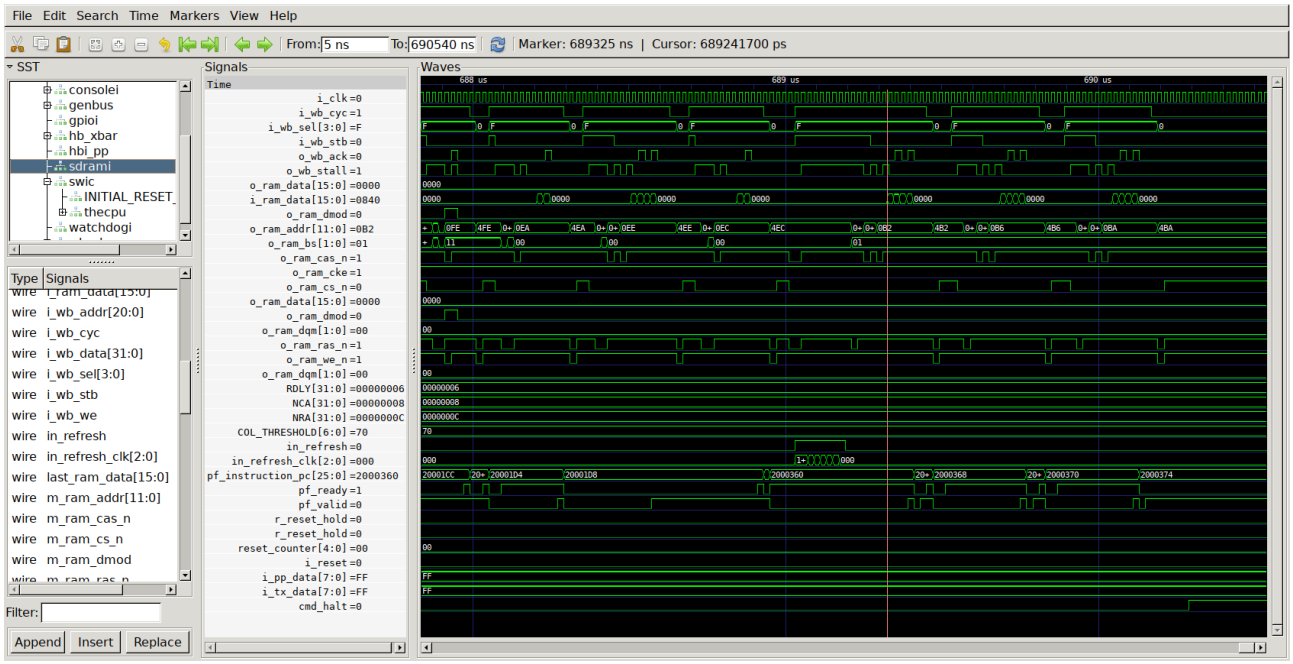
20001d4: 87 fa fc f8 JSR 0x02000360 // 2000360 <puts>
20001d8: 02 00 03 60

fig8.



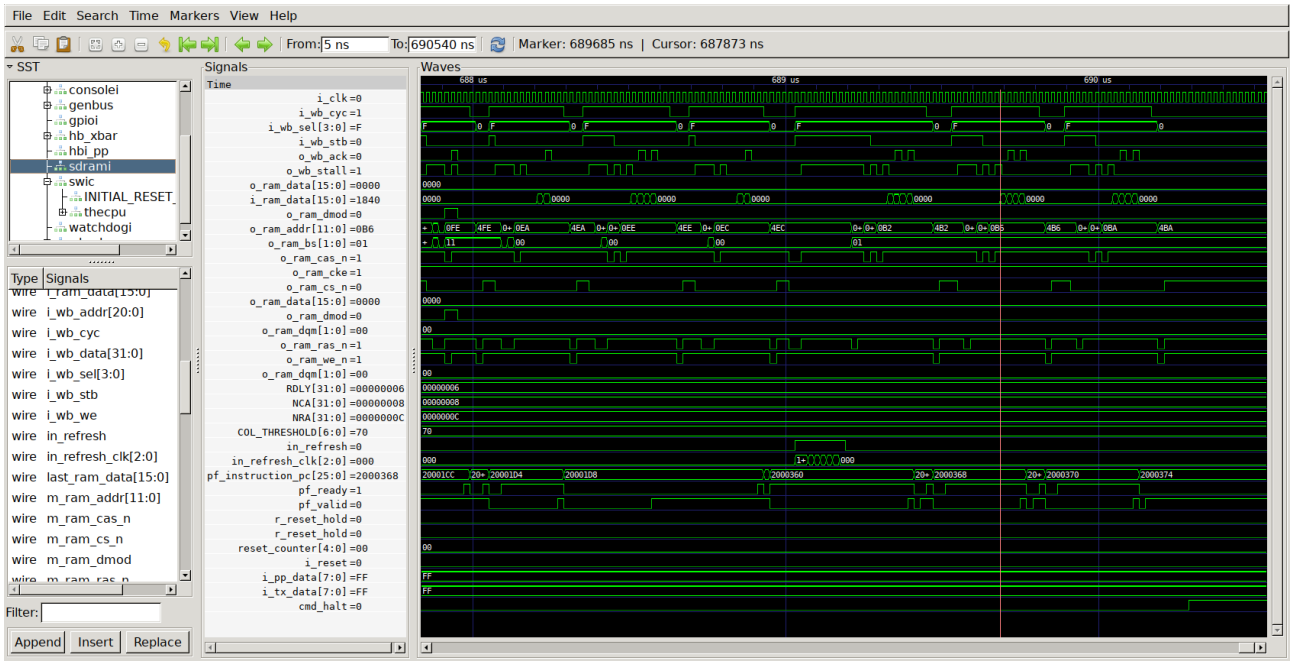
02000360 <puts>:
2000360: e8 28 85 1c SUB $40,SP | SW R0,$28(SP)

fig9.



2000368: 2b 40 40 00 MOV R1,R5
200036c: 12 03 00 40 LDI 0x0200e944,R2 // 200e944 <_impure_ptr>

fig10.



2000370: 12 40 e9 44
2000374: 34 84 80 00 LW (R2),R6

fig11.

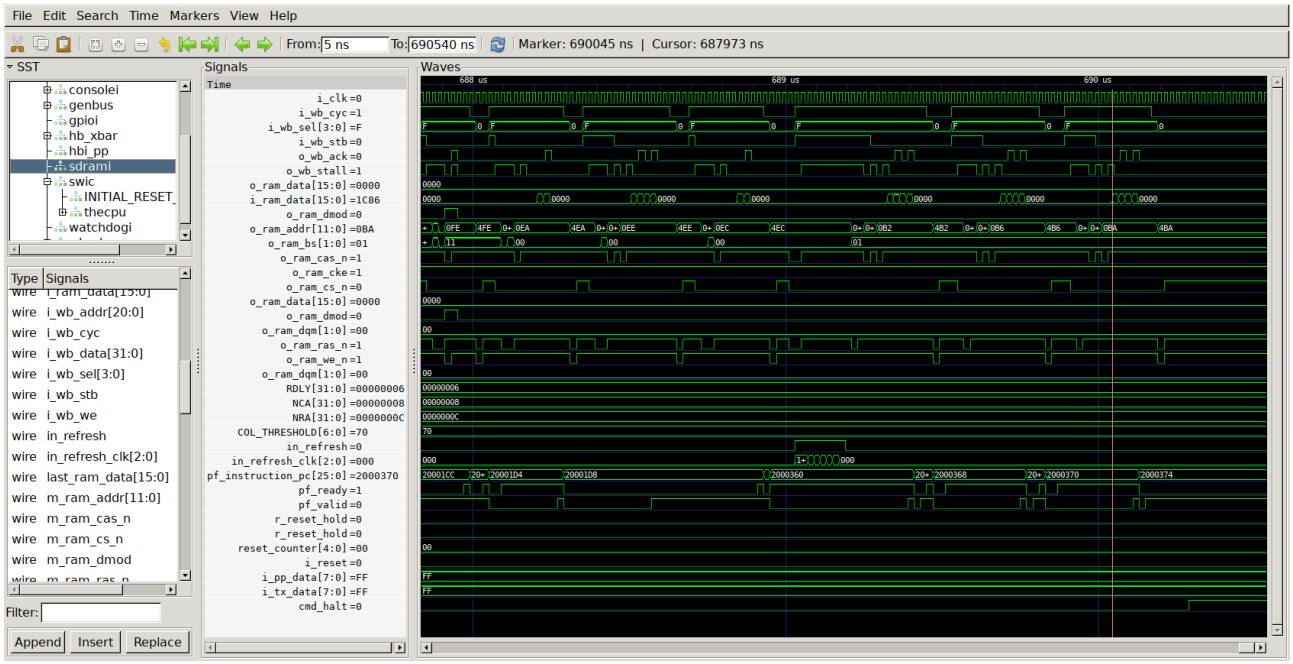
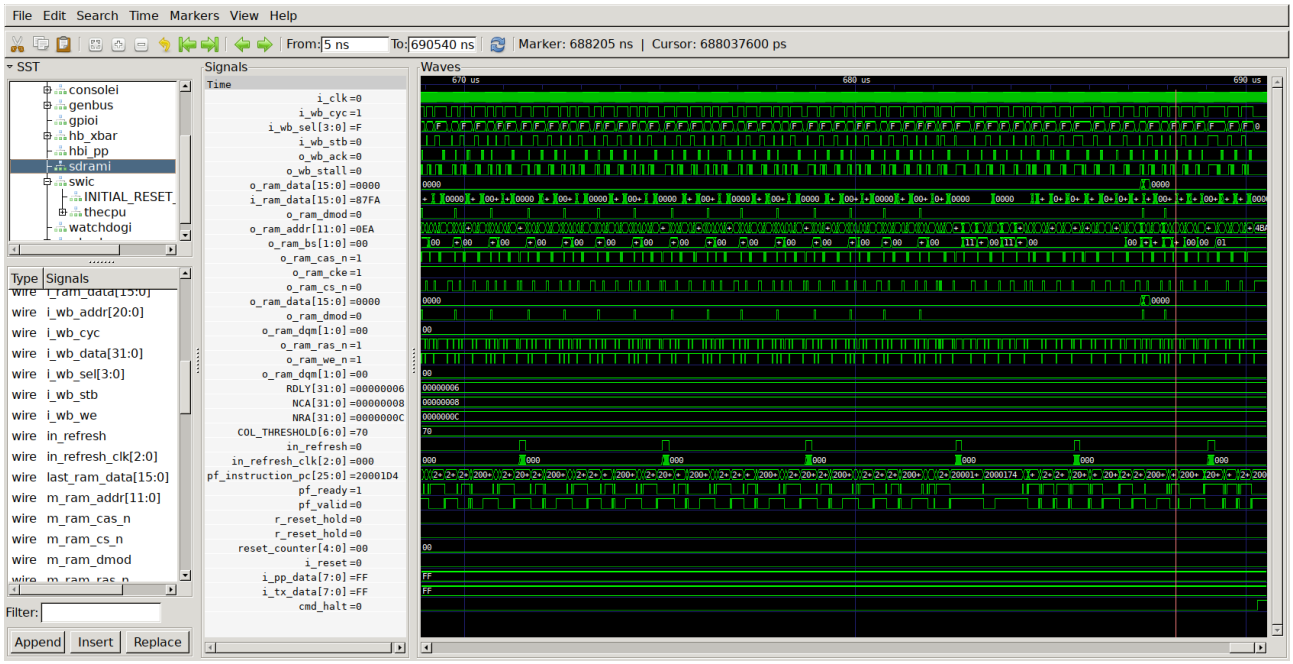


Fig12.



cmd_halt goes hi.

Below is how the program should work.

/pc-main_tb ../../sw/board/hello
Listening on port 8363
Listening on port 8364
> T
Hello, World!
w & h were set
w=256 h=256

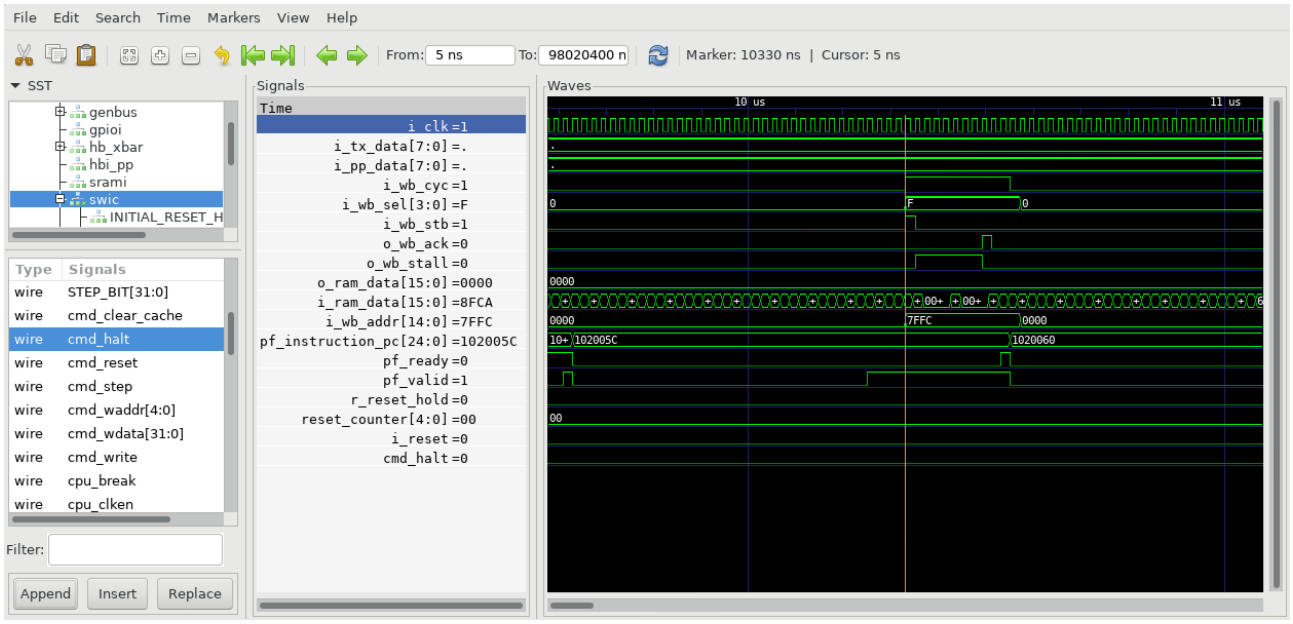
BOMB : CPU BREAK RECEIVED
ZIPM--DUMP: Supervisor mode

sR0 : 00e00060 sR1 : 0000000d sR2 : 00000000 sR3 : 00800000
sR4 : 0000000a sR5 : 00e0f10c sR6 : 00000000 sR7 : 00000000
sR8 : 00000000 sR9 : 00000000 sR10: 00000000 sR11: 00000000
sR12: 00000000 sSP : 00e1ffec sCC : 00000000 sPC : 00e00060

uR0 : 00000000 uR1 : 00000000 uR2 : 00000000 uR3 : 00000000
uR4 : 00000000 uR5 : 00000000 uR6 : 00000000 uR7 : 00000000
uR8 : 00000000 uR9 : 00000000 uR10: 00000000 uR11: 00000000
uR12: 00000000 uSP : 00000000 uCC : 00000020 uPC : 01020048

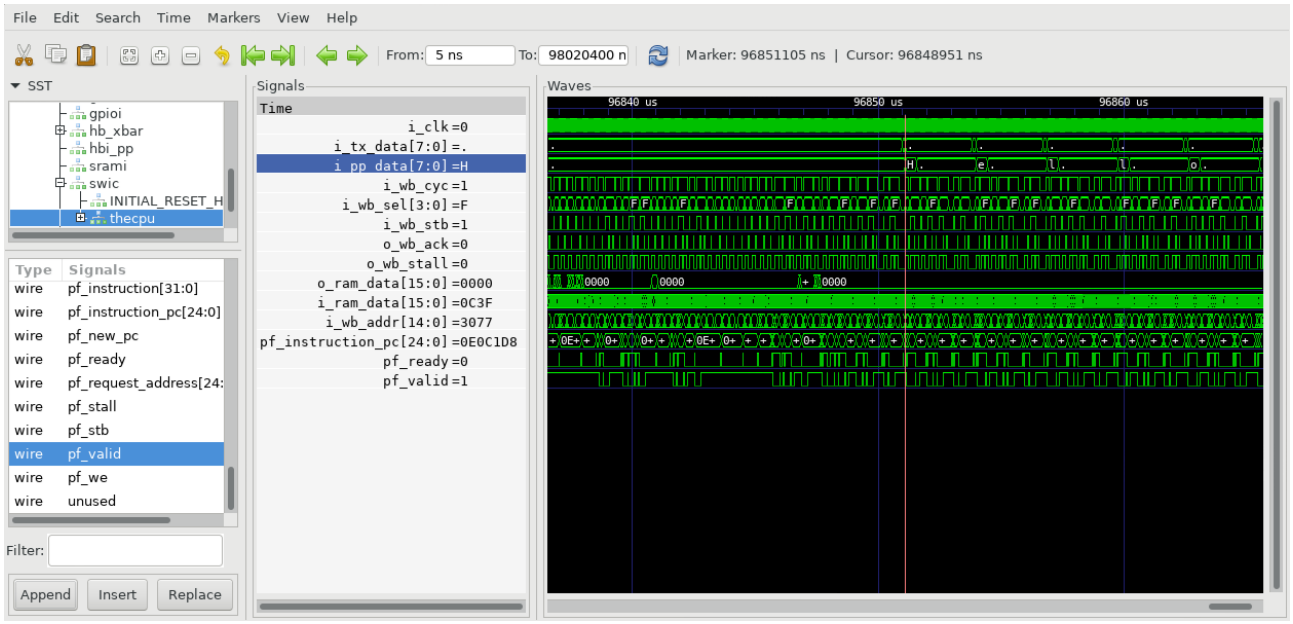
Will exit: DONE!!

fig13.



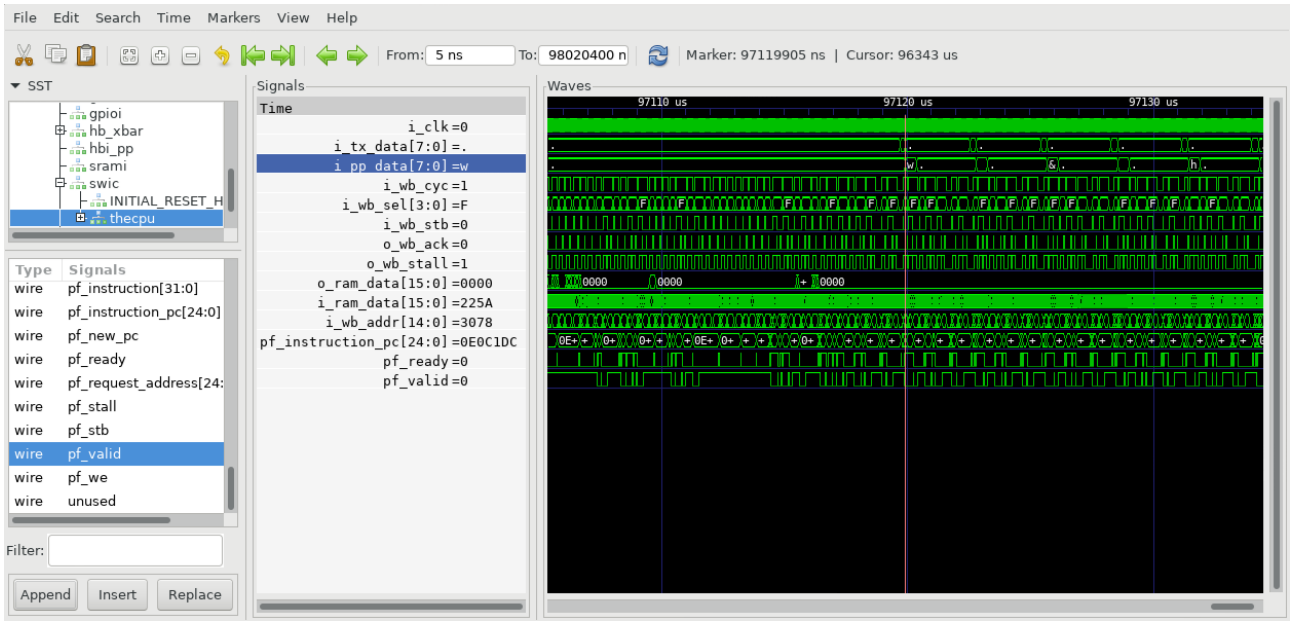
Sending string “Hello, World!”

Fig14.



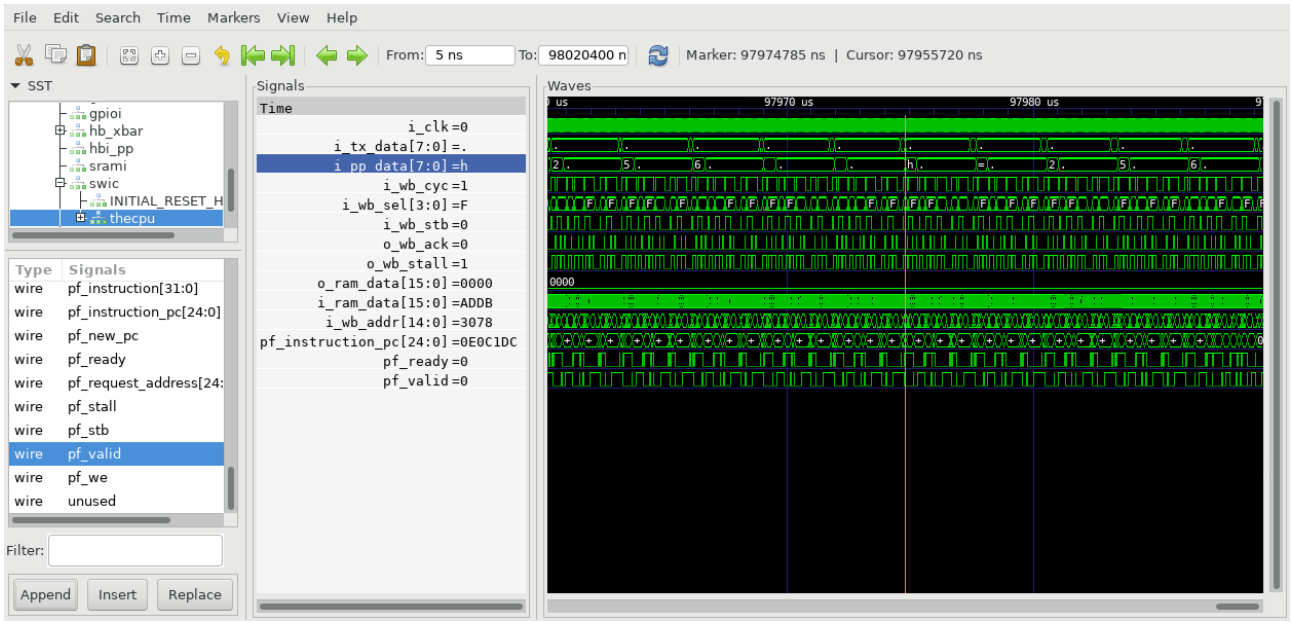
Sending string “w & h were set”

fig15.



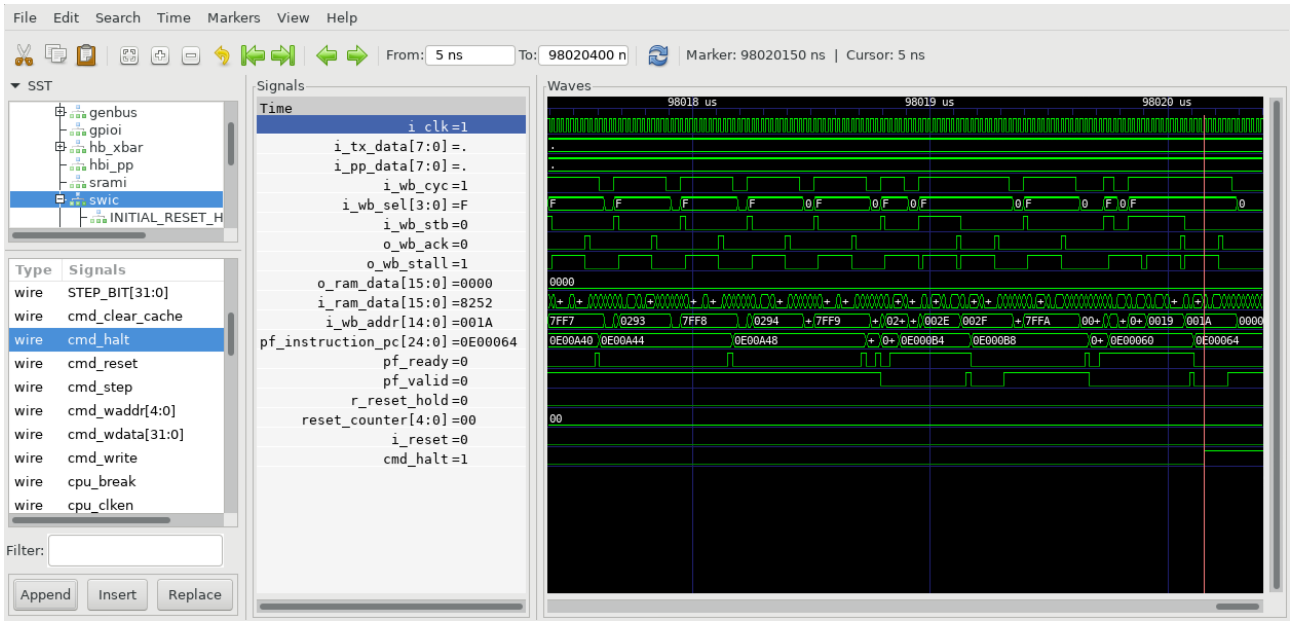
Sending string”w=256 h=256”

fig16.



cmd_halt goes hi.

Fig17.



hello.c

```
////////////////////////////////////////////////////////////////////////////////
//
// Filename:    hello.c
// {{{
// Project:     ICO Zip, iCE40 ZipCPU demonstration project
//
// Purpose:     The original Helllo World program.  If everything works, this
//              will print Hello World to the UART, and then halt the CPU--if
//      run with no O/S.
//
//
```

```
////////////////////////////////////////////////////////////////////////////////
//
// Gisselquist Technology asserts no ownership rights over this particular
// hello world program.
//
////////////////////////////////////////////////////////////////////////////////
//
// }}}
#include <stdio.h>

#include <stdlib.h>
#include <zipcpu.h>

#define BLKRAM_FLAG 0x01401000
#define BLKRAM_INVFWD 0x01401004
#define BLKRAM_WAIT 0x01401008
#define BLKRAM_INP 0x0140100c

#define BLKRAM_WAIT1 0x01401010
#define BLKRAM_INP1 0x014010c0

#define BLKRAM_WAIT2 0x01401018
//#define BLKRAM_INP2 0x0140101c

#define imgsize 256
#define DBUG 1
#define DBUG1 1

struct PTRs {
        int inpbuf[256];
        int flag;
        int wait;
        int wait1;
        int wait2;
        int w;
        int h;
        /*
          ptrs.red = ( int *)malloc(sizeof( int)* ptrs.w*ptrs.h*2);
          first 65536 used as input to lifting
          2nd 655536 used as output for lifting.
        */
        int *red;
        int *alt;
        int *ptr_blkram_flag;
        int *ptr_blkram_invfwd;
        int *ptr_blkram_wait;
        int *ptr_blkram_inp;
        int *ptr_blkram_wait1;
        int *ptr_blkram_wait2;
        int *ptr_blkram_inp1;

} ptrs;
```

```
int main(int argc, char **argv) {
	printf("Hello, World!\n");

	ptrs.w = 256;
	ptrs.h = 256;
	ptrs.ptr_blkram_inp = (int *)BLKRAM_INP;
	*ptrs.ptr_blkram_inp = &(ptrs.w);
		ptrs.ptr_blkram_inp1 = (int *)BLKRAM_INP1;
	*ptrs.ptr_blkram_inp1 = &(ptrs.h);
	printf("w & h were set\n");
	printf("w=%d  h=%d\n",ptrs.w,ptrs.h);

	zip_break();


}
```