SOC Design Lab 4-1

Group 8

- Firmware code

1.fir.h

Define the taps numbers and input data

```
#ifndef __FIR_H__
#define __FIR_H__

#define N 11

int taps[N] = {0,-10,-9,23,56,63,56,23,-9,-10,0};
int inputbuffer[N];
int inputsignal[N] = {1,2,3,4,5,6,7,8,9,10,11};
int outputsignal[N];
#endif
```

2. fir.c

1) void initfir()

This is to reset the input and output buffers before doing fir conputing.

```
#include "fir.h"

void __attribute__ ( ( section ( ".mprjram" ) ) ) initfir() {
    for (int i = 0; i < N; i++) {
        inputbuffer[i] = inputsignal[i];
    }
    for (int i = 0; i < N; i++) {
        outputsignal[i] = 0;
    }
}
```

2) Int* fir()

Using a for loop to compute the fir and save the result to “outputsignal”.

Return the pointer of the “outputsignal”.

```
int* __attribute__ ( ( section ( ".mprjram" ) ) ) fir(){
    initfir();
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            if (i-j >= 0) {
                outputsignal[i] += inputbuffer[i-j] * taps[j];
            }
        }
    }
    return outputsignal;
}
```

- # Interface between BRAM and wishbone

1. To make sure the mprjram is selected ,where wbs_adr_i[31:20] is from 12’h380 (12’b0011_1000_0000) to 12’h384 (12’b0011_1000_0100) , we assign a signal Sel as wbs_adr_i[31:23] is 9’h070 (9’b001110000) to determine whether the wishbone address is in the range.
2. clock (clk) of the BRAM and reset (rst) are assigned to the ones from wishbone (wb_clk_i and wb_rst_i)
3. enble (EN0) of the BRAM is assigned to wbs_cyc_i && wbs_stb_i && Sel
4. write enable (WE0) is assigned to wbs_sel_i & {4{wbs_we_i}}
5. DataIn(write data) is assigned to wbs_dat_i
6. Address(A0) is assigned to Sel ? (wbs_adr_i & 32'h003FFFFF) : 32’b0, using a mask to make the bram size smaller.
7. We used a simple 4 bits counter and a “ready” signal to control the desired delay time from enable to wbs_ack_o.

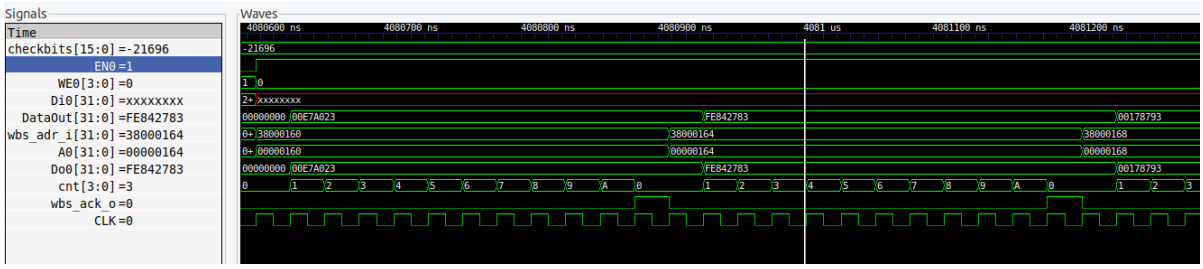
```
assign wbs_ack_o = ready;

always @(posedge clk) begin
    if (rst ) begin
        ready <= 0;
    end else if (EN0 && !ready) begin
        ready <= 0;
        if (cnt == DELAYS) begin
            cnt <= 4'b0;
            ready <= 1'b1;
        end else cnt <= cnt + 1;
    end else begin
        cnt <= 0;
        ready <= 0;
    end
end
```

8. DataOut(read data) is connected to wbs_data_o for output.
9. BRAM size parameter N = 9.

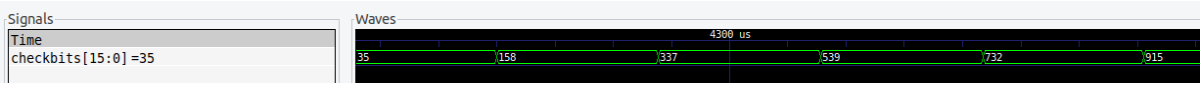
```
parameter N = 9;
(* ram_style = "block" *) reg [31:0] RAM[0:2**N-1];
```

- # Waveforms

1. The communications between CPU and BRAM is shown below:



2. FIR results can be checked from the checkbits of the testbench.



- Synthesis report

RTL Component:

```
---------------------------------------------------------------------
Start RTL Component Statistics
---------------------------------------------------------------------
Detailed RTL Component Info :
+---Adders :
      2 Input    4 Bit       Adders := 1
+---Registers :
                  32 Bit    Registers := 1
                   4 Bit    Registers := 1
                   1 Bit    Registers := 1
+---RAMs :
                 16K Bit   (512 X 32 bit)          RAMs := 1
+---Muxes :
      2 Input   32 Bit        Muxes := 7
      2 Input    8 Bit        Muxes := 1
      2 Input    4 Bit        Muxes := 1
      2 Input    1 Bit        Muxes := 3
---------------------------------------------------------------------
Finished RTL Component Statistics
```

Slice Logic:

```
1. Slice Logic
--------------

+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs*             |   34 |     0 |          0 |     53200 |  0.06 |
|   LUT as Logic          |   34 |     0 |          0 |     53200 |  0.06 |
|   LUT as Memory         |    0 |     0 |          0 |     17400 |  0.00 |
| Slice Registers         |    5 |     0 |          0 |    106400 | <0.01 |
|   Register as Flip Flop |    5 |     0 |          0 |    106400 | <0.01 |
|   Register as Latch     |    0 |     0 |          0 |    106400 |  0.00 |
| F7 Muxes                |    0 |     0 |          0 |     26600 |  0.00 |
| F8 Muxes                |    0 |     0 |          0 |     13300 |  0.00 |
+-------------------------+------+-------+------------+-----------+-------+
```

Memory:

```
2. Memory
---------

+-------------------+------+-------+------------+-----------+-------+
|     Site Type     | Used | Fixed | Prohibited | Available | Util% |
+-------------------+------+-------+------------+-----------+-------+
| Block RAM Tile    |  0.5 |     0 |          0 |       140 |  0.36 |
|   RAMB36/FIFO*    |    0 |     0 |          0 |       140 |  0.00 |
|   RAMB18          |    1 |     0 |          0 |       280 |  0.36 |
|     RAMB18E1 only |    1 |       |            |           |       |
+-------------------+------+-------+------------+-----------+-------+
```