reg [0: C\_S\_AXI\_DATA\_WIDTH-1] tmp\_reg;

always @( posedge S\_AXI\_ACLK ) begin // on rising clock edge

if ( S\_AXI\_ARESETN == 1'b0 ) begin // when reset is low

slv\_reg2 <= 0;

tmp\_reg <= 0;

end

else begin // reset is high, multiply the values of the reg0, reg 1

tmp\_reg <= slv\_reg0 \* slv\_reg1;

slv\_reg2 <= tmp\_reg; // store in reg2

end

end

#include <stdio.h>

#include "platform.h"

#include "xil\_printf.h"

#include <xparameters.h>

#include <multiply.h>

#include "xil\_io.h"

#include "xil\_types.h"

int main() {

init\_platform();

int ans;

print("Hello World\n\r");

for (int i = 0; i < 16; ++i) {

MULTIPLY\_mWriteReg(XPAR\_MULTIPLY\_0\_S00\_AXI\_BASEADDR, 0, i);

printf("siv\_reg0 = %d \n", i);

MULTIPLY\_mWriteReg(XPAR\_MULTIPLY\_0\_S00\_AXI\_BASEADDR, 4, i);

printf("siv\_reg1 = %d \n", i);

ans = MULTIPLY\_mReadReg(XPAR\_MULTIPLY\_0\_S00\_AXI\_BASEADDR, 8);

printf("ans = %d \* %d = %d \n\n", i, i, ans);

}

cleanup platform();

return 0;

}