COL 216 Assignment 2 Stage 5

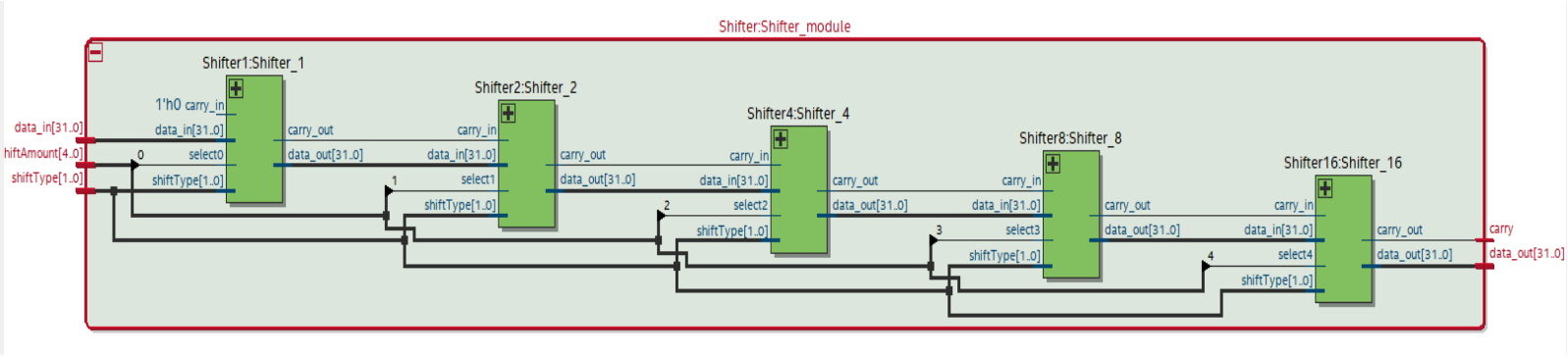
Chinmay Mittal (2020CS10336)

Implement the Shifter Module in the multicycle design

New components added

1. Shifter: The main entity which is instantiated in the processor file
2. Shifter1: subcomponent to shift by 1 bit
3. Shifter2: subcomponent to shift by 2 bits
4. Shifter4: subcomponent to shift by 4 bits
5. Shifter8: subcomponent to shift by 8 bits
6. Shifter16: subcomponent to shift by 16 bits

I have used inbuilt *shift_right* and *shift_left* functions from *IEEE.numeric_std*
Synthesis output of Shifter

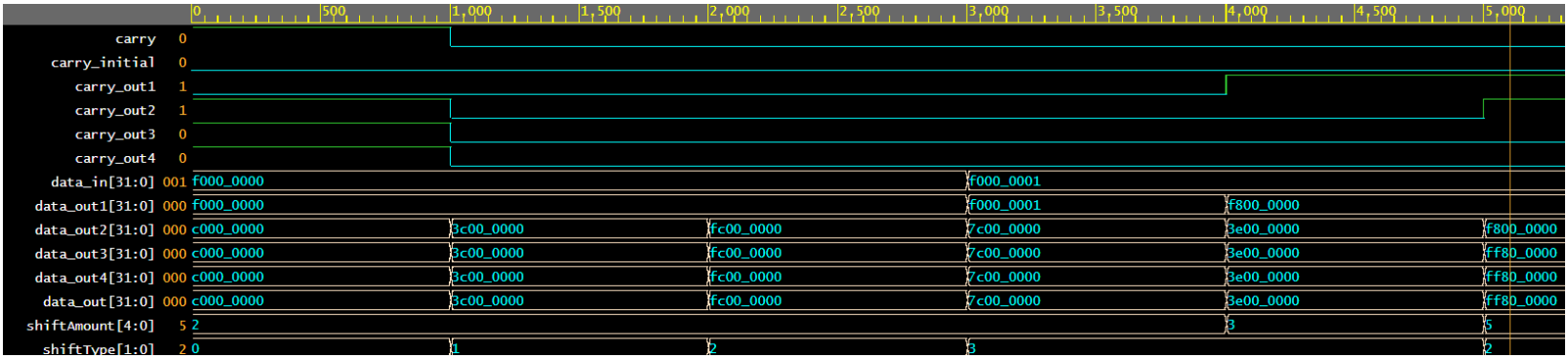


New states in the FSM (for shifting and reading additional registers), additional multiplexing logic have been added and additional functionality for reading offset of DT instructions from registers which were previously omitted has now been added.

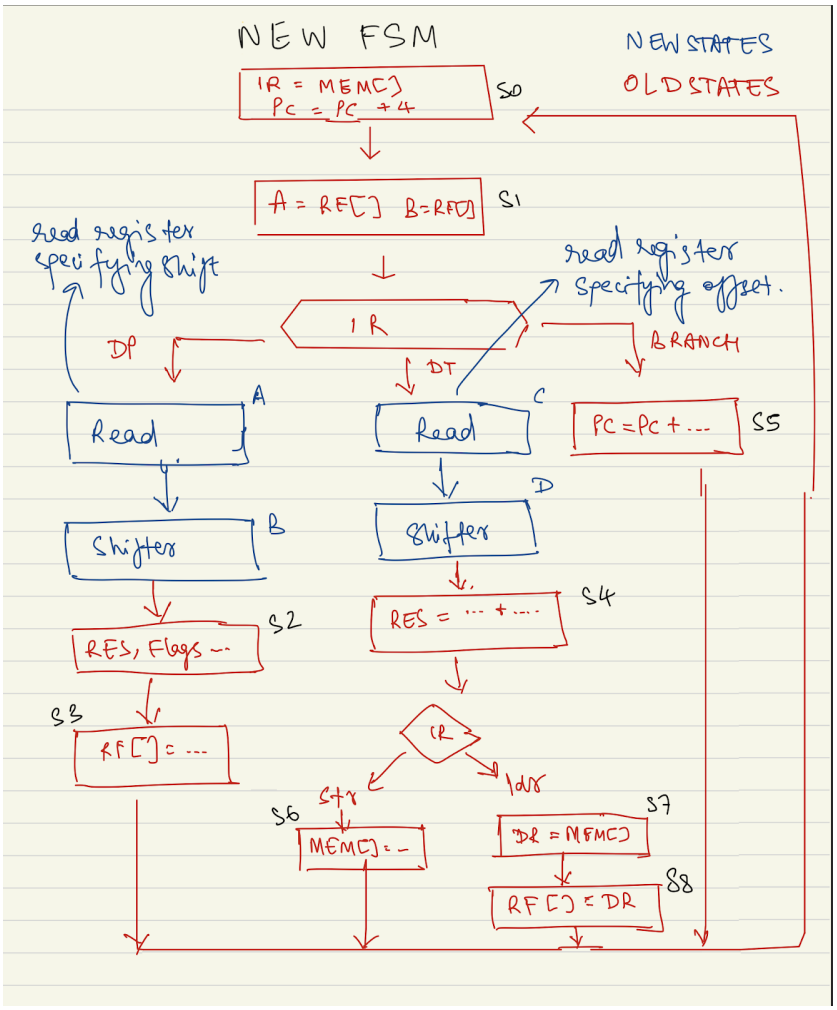
The following shifts are implemented (as in assignment description) :

1. DP
    a. If the operand is constant then ROR can be performed by bits in the instruction
    b. If the operand is a register then
        i. Shift can be specified by a constant
        ii. Shift can be read from a register
2. DT
    a. If offset is a constant then no shift is done
    b. If the offset is read from a register, then the shift can specified by a constant amount specified within the instruction (this shift will not be from a register as specified in the assignment)

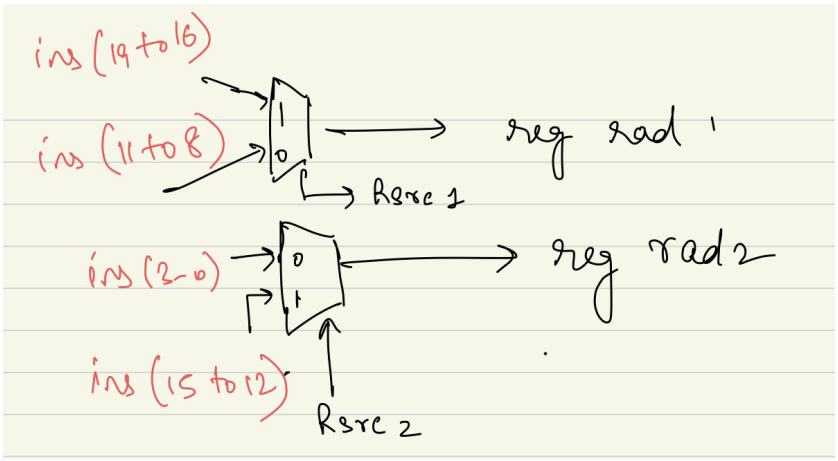
I have created a testbench for testing the new Shifter component
EP Wave for this testbench is as follows:

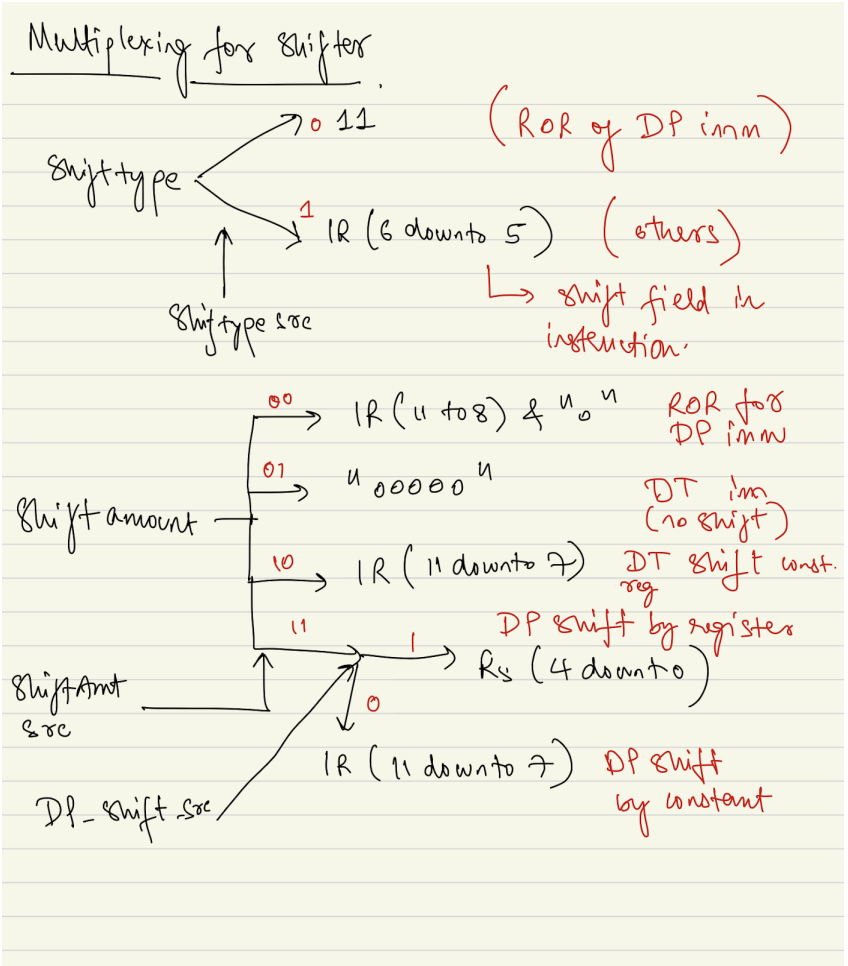


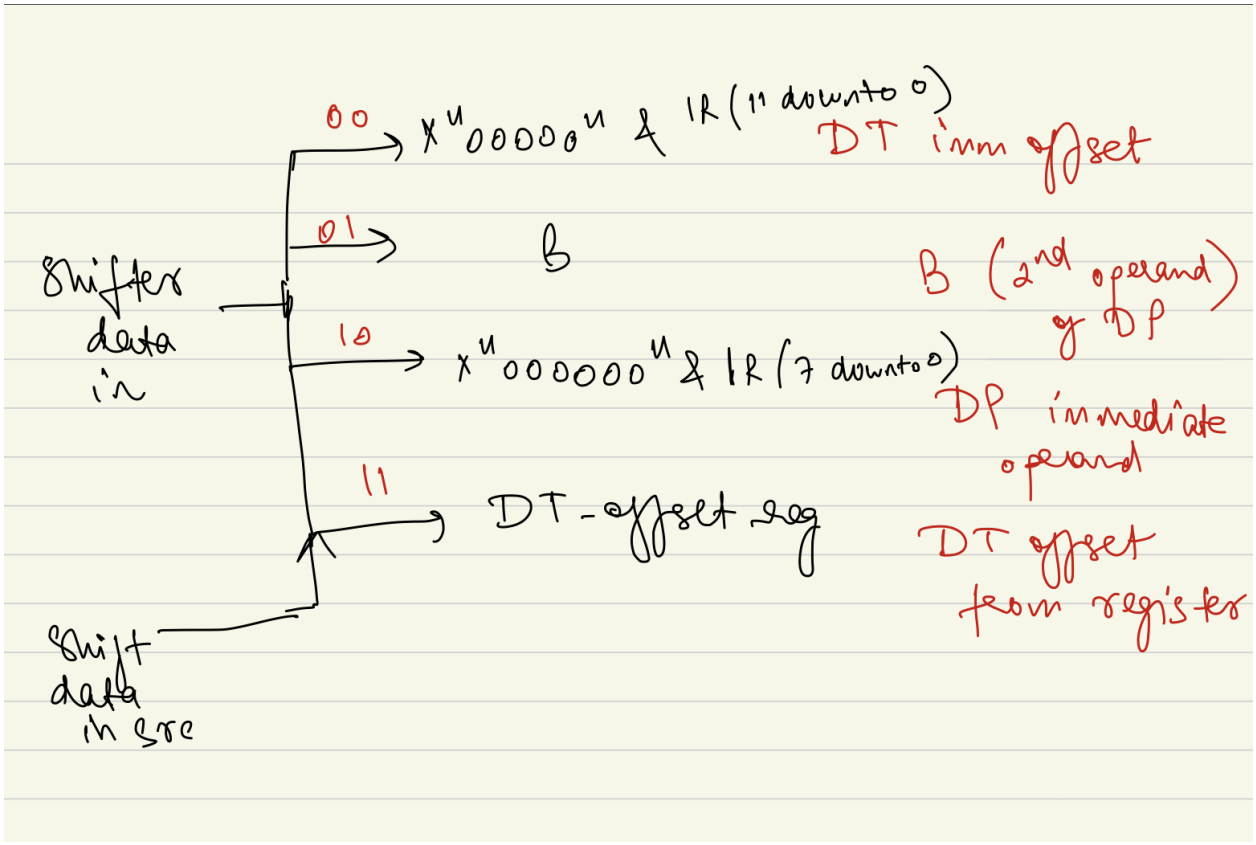
The FSM is slightly modified to dedicate states to the shifter and reading additional registers required for DP or DT instructions.



Since additional registers have to be read additional multiplexing has been provided for the register file and the names of the control signals have been changed slightly. Multiplexing for the shifter is also shown.


ins (19 to 16)
ins (11 to 8)
1
0
Rsrc 1
reg rad 1
ins (3-0)
0
1
reg rad 2
ins (15 to 12)
Rsrc 2


Multiplexing for shifter
shift type
0
11
(ROR of DP imm)
1
IR (6 downto 5)
(others)
shift field in instruction.
shift type src
00
IR (11 to 8) & "0"
ROR for DP imm
01
"00000"
DT imm (no shift)
shift amount
10
IR (11 downto 7)
DT shift const.
reg
11
DP shift by register
1
Rs (4 downto 0)
ShiftAmt src
0
IR (11 downto 7)
DP shift by constant
DP_shift src

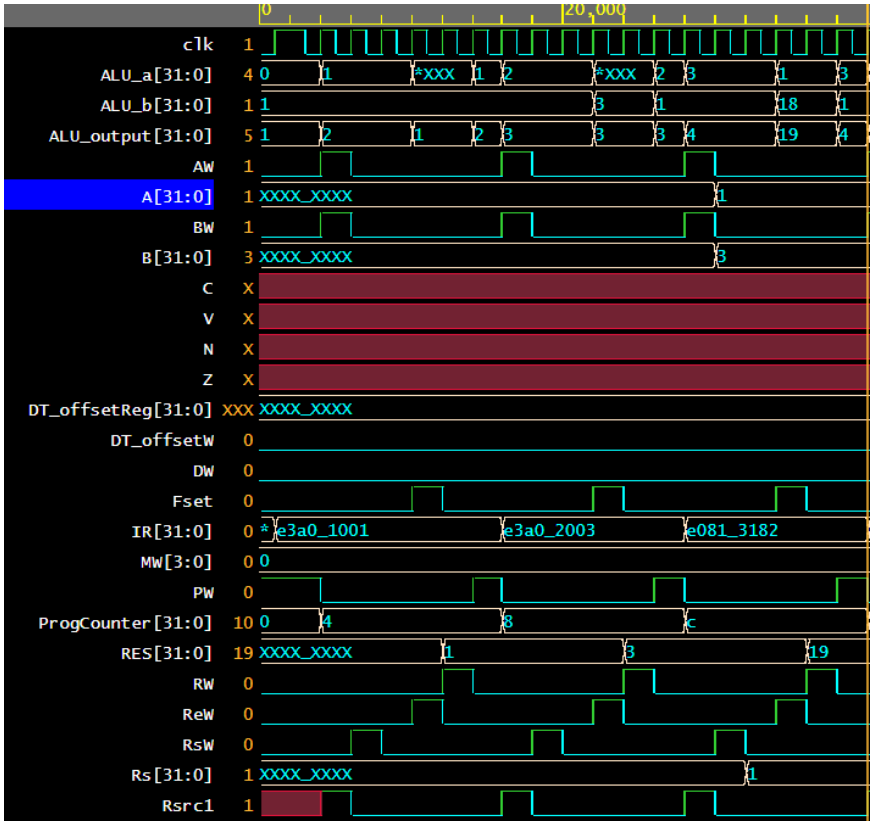


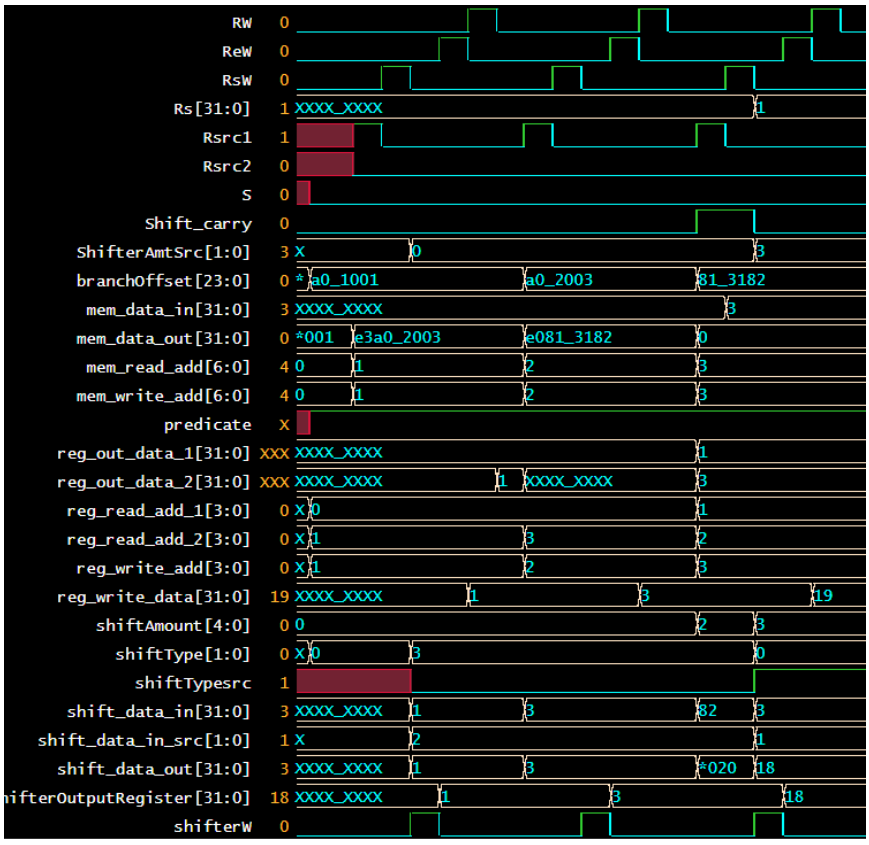
I have added some additional registers for the Multicycle datapath and their control signals,

1. Rs stores the shift amount specified by a register in case of DP instructions
2. DT_offsetReg stores the offset value read from the register file for DT instructions
3. shifterOutputRegister, the clocked output of the shifter component

Test programs
I have added assembly code for test programs in the submission

Test1.s
EP WAVE





Test2.s
EP WAVE

Test3.s
EP WAVE