COL 216 Assignment 2 Stage 5

Manshi Sagar - 2020CS50429

March 2022

1 Introduction

In this stage, we aim to incorporate the shift and reduce features in DP and DT instructions. For this, we design a Shift Rotate Unit which is positioned between register file and ALU.

2 Shift Rotate Unit - SRU

Inputs are:

- clock
- shifttype: Type of shift : ROR or LSR or LSL or ASR
- shift-amount:
- val: value that needs to be shifted

Outputs are:

- shiftedval: value obtained after shifting
- shift-carry: carry obtained after shifting

To obtains shifted result in one clock cycle and using combinational circuit, we design 5 helper components to perform n-bit shifts(n=1 to 5).

- sru1: input-val, output-shiftedval1
- sru1: input-shiftedval1, output-shiftedval2
- sru1: input-shiftedval2, output-shiftedval3
- sru1: input-shiftedval3, output-shiftedval4
- sru1: input-shiftedval4, output-shiftedval5

Final output of SRU = shiftedval5
This shifted value is given into the ALU as operand2.

3 Finite State Machine

New states, Registers, control signals and multiplexers are introduced:

3.1 Registers

- SA: to store shift amount
- SV: to store shift value
- SRES: to store shifted value from SRU

3.2 Control Signals

- SAW: write enable for SA
- SVW: write enable for SV
- SREW: write enable for SRES

3.3 Multiplexers

- RSCR1: to chose read address1 of register file

```
if(RSRC1 = '1') then
    rad1<= ir(11 downto 8);
else rad1<= ir(19 downto 16);
end if ;
```

- SAM: to chose Shift Amount before storing in SA

```
if(SAM = "00") then
    shift_amount<= rd1(4 downto 0);
    elsif(SAM="01") then
    shift_amount<= ir(11 downto 7);
    elsif(SAM="10") then
    shift_amount<= ir(11 downto 8)&'0';
    else
    shift_amount<= "00000";
end if;
```

- SVM: to chose Shift Value before storing in SV

```
if(SVM = "00") then
    val<=X"000000" & IR(7 downto 0);
elsif(SVM ="01") then
    val<=rd2;
elsif(SVM<="10") then
    val<=X"00000" & IR(11 downto 0);
end if;
```

3.4 States

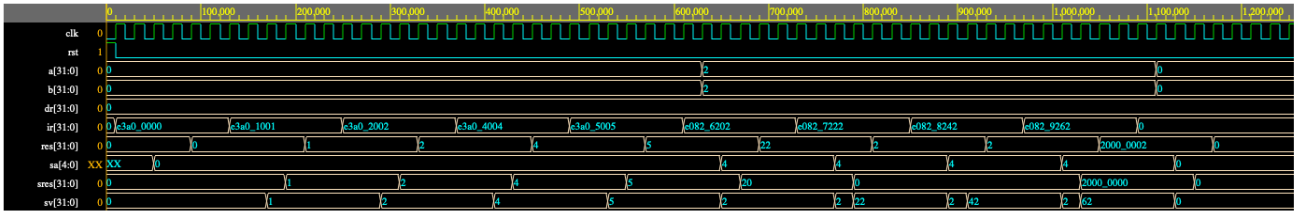
- State10 - to write appropriate value in SA and SV
- State11 - to store shifted value in SRES

3.5 Synthesis result

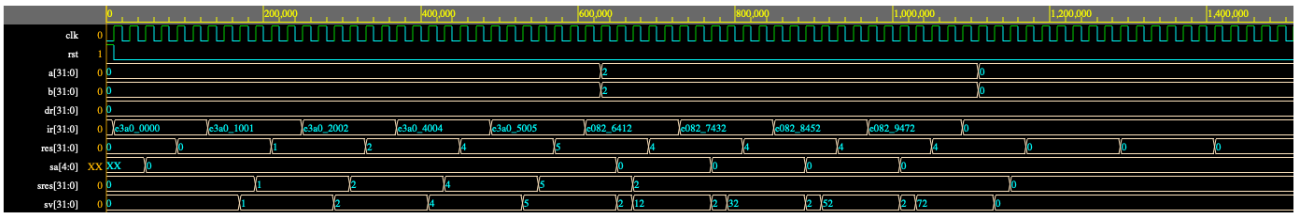
```
# Info: ************************************************************
# Info: Device Utilization for 7A100TCSG324
# Info: ************************************************************
# Info: Resource                          Used    Avail   Utilization
# Info: ------------------------------------------------------------
# Info: IOs                               73      210      34.76%
# Info: Global Buffers                    1       32        3.12%
# Info: LUTs                              179     63400     0.28%
# Info: CLB Slices                        40      15850     0.25%
# Info: Dffs or Latches                   33      126800    0.03%
# Info: Block RAMs                        0       135       0.00%
# Info: DSP48E1s                          0       240       0.00%
# Info: ------------------------------------------------------------
# Info: ************************************************
```

4 TestCases

Assembly code	ARM Big Endian
mov r0,#0	E3A00000
mov r1,#1	E3A01001
mov r2,#2	E3A02002
mov r4,#4	E3A04004
mov r5,#5	E3A05005
add r6,r2,r2, LSL #4	E0826202
add r7,r2,r2, LSR #4	E0827222
add r8,r2,r2 ,ASR #4	E0828242
add r9,r2,r2, ROR #4	E0829262



Assembly code	ARM Big Endian
mov r0,#0	E3A00000
mov r1,#1	E3A01001
mov r2,#2	E3A02002
mov r4,#4	E3A04004
mov r5,#5	E3A05005
add r6,r2,r2, LSL r4	E0826412
add r7,r2,r2, LSR r4	E0827432
add r8,r2,r2, ASR r4	E0828452
add r9,r2,r2, ROR r4	E0829472



Assembly code	ARM Big Endian
mov r0,#0	E3A00000
mov r1,#1	E3A01001
mov r2,#2	E3A02002
mov r4,#4	E3A04004
mov r5,#5	E3A05005
str r6, [r1, r4, LSR #4]	E7816224
ldr r8, [r0, r4, ASR #4]	E7908244