VLSI Final Exam

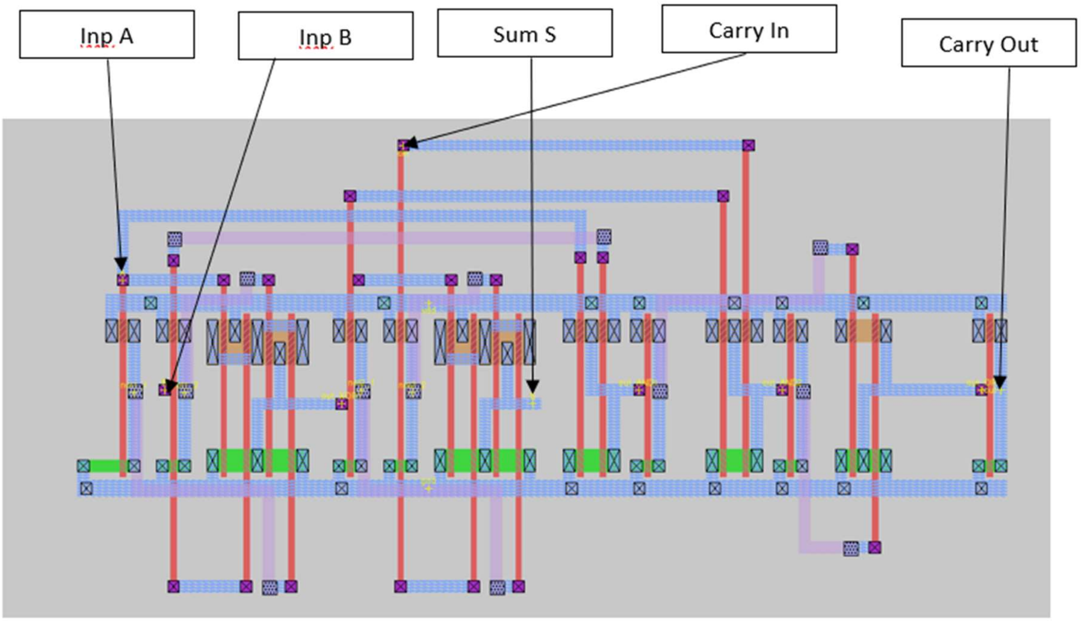
Raghavendra Chathurajupalli

All related files including mag, spice ext and spice text files are attached in the submission under different folder each corresponding to different questions.

1.

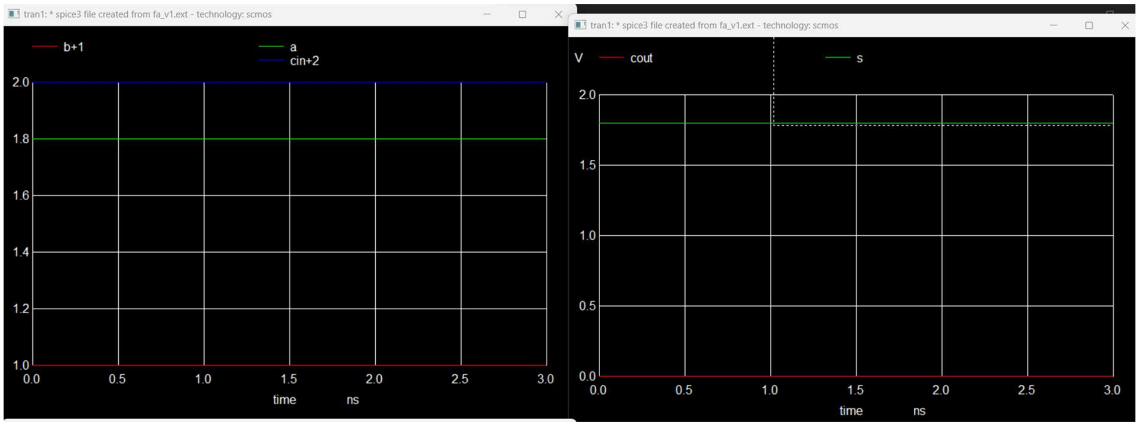
Draw a 1-bit full adder (FA) circuit using MAGIC, make sure it passes all DRCs, extract the circuit level netlist, and simulate its behaviour using SPICE. (10 points)

Magic Output:

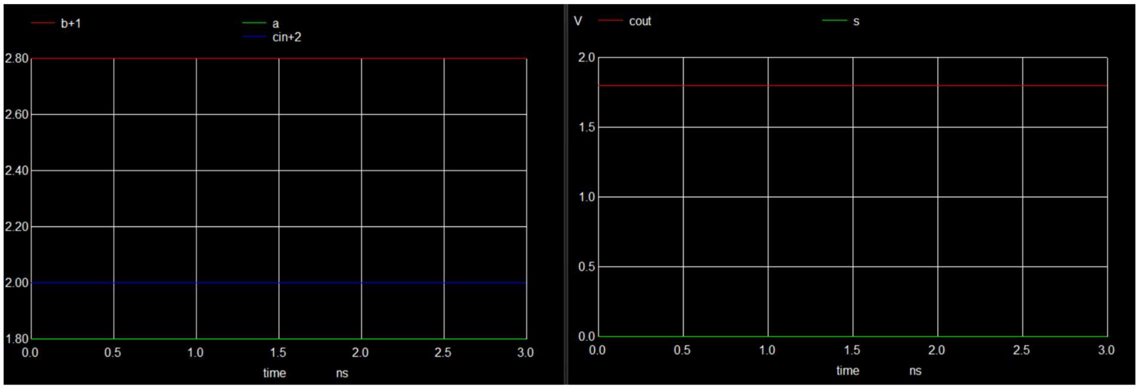


NGSPICE Simulation (spice txt file attached with submission):

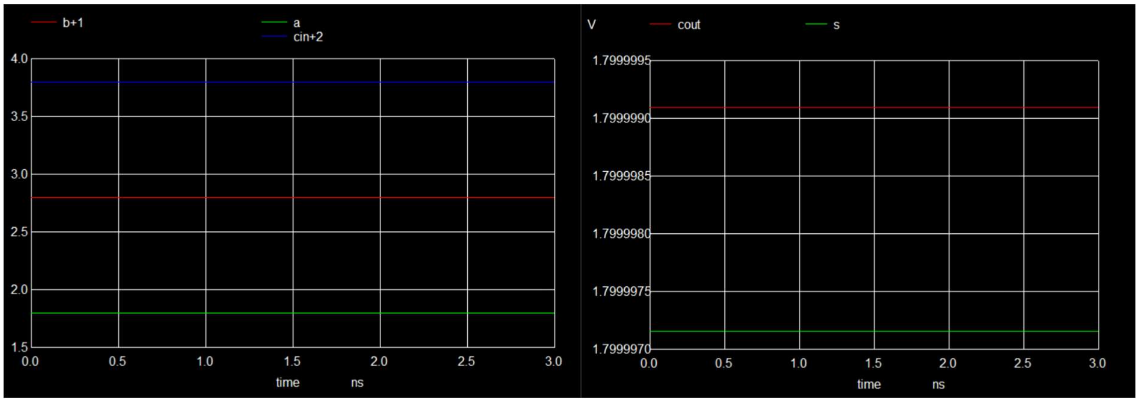
A=1.8, B=Cin=0, S=1.8, Cout=0



A=B=1.8 Cin=0, S=0, Cout=1



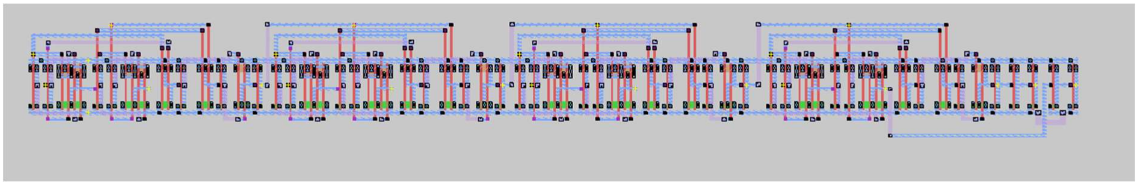
A=B=Cin=1.8, S=Cout=1.79 (approx. 1.8)



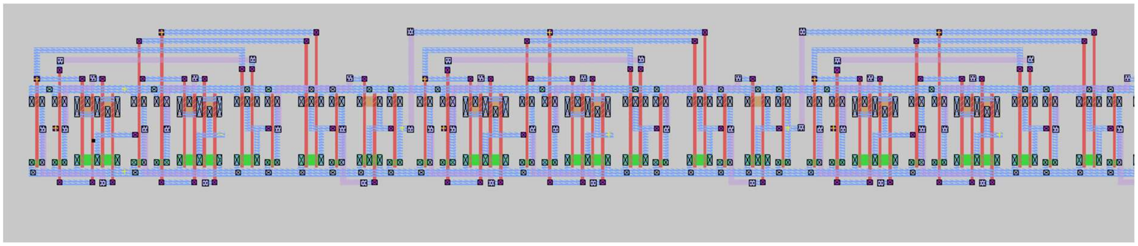
2. **Use hierarchical design flow to instantiate four units of FA to create a 4-bit ripple carry adder (RCA), make sure it passes all DRCs, extract the circuit level netlist, and simulate its behaviour using SPICE with (a) 4+5 and (b) 4-5 examples using two's complement representation. (20 points)**

   Magic Design:

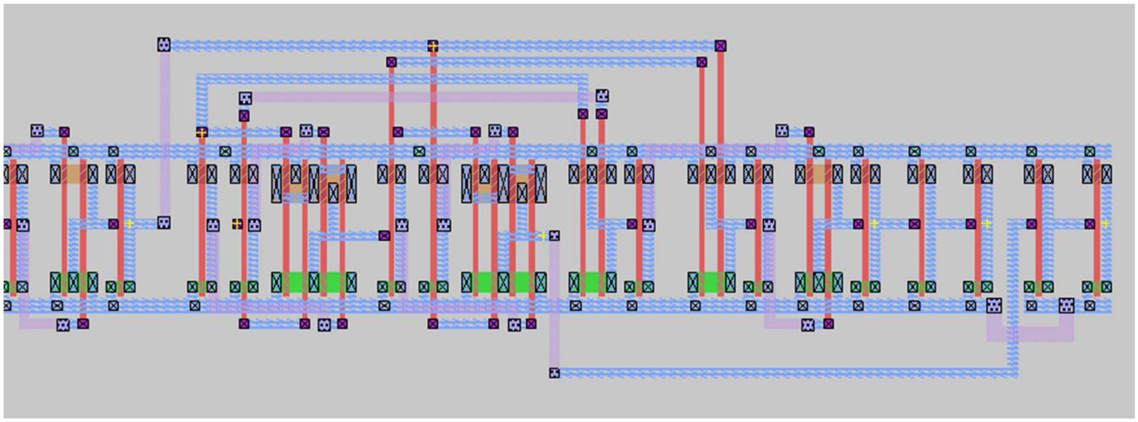
   High level View:



   Left side:



   Right side:

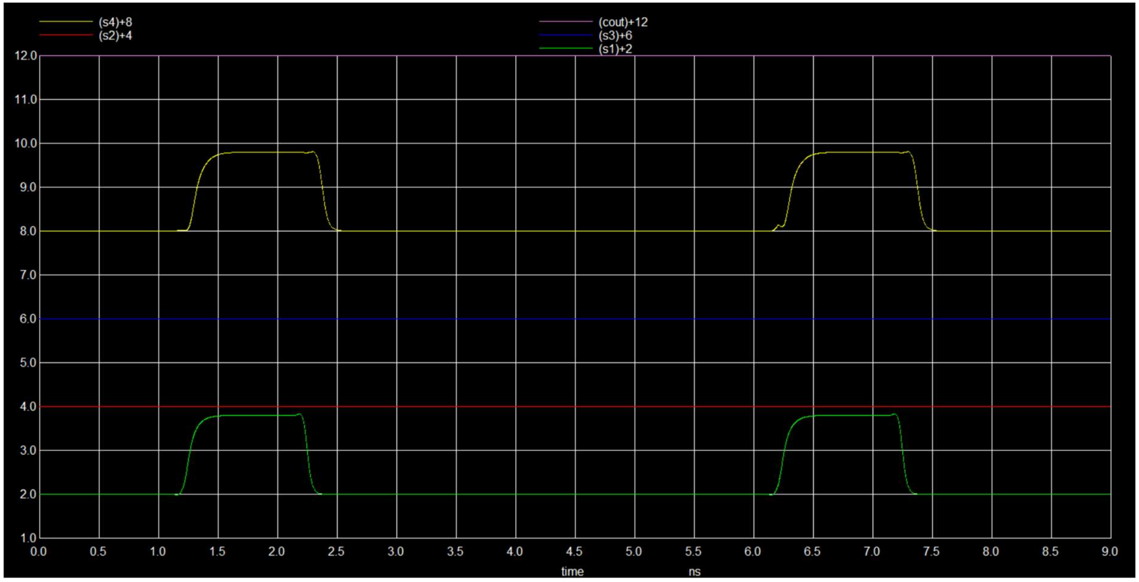


a) 4+5=0100+0101=1001

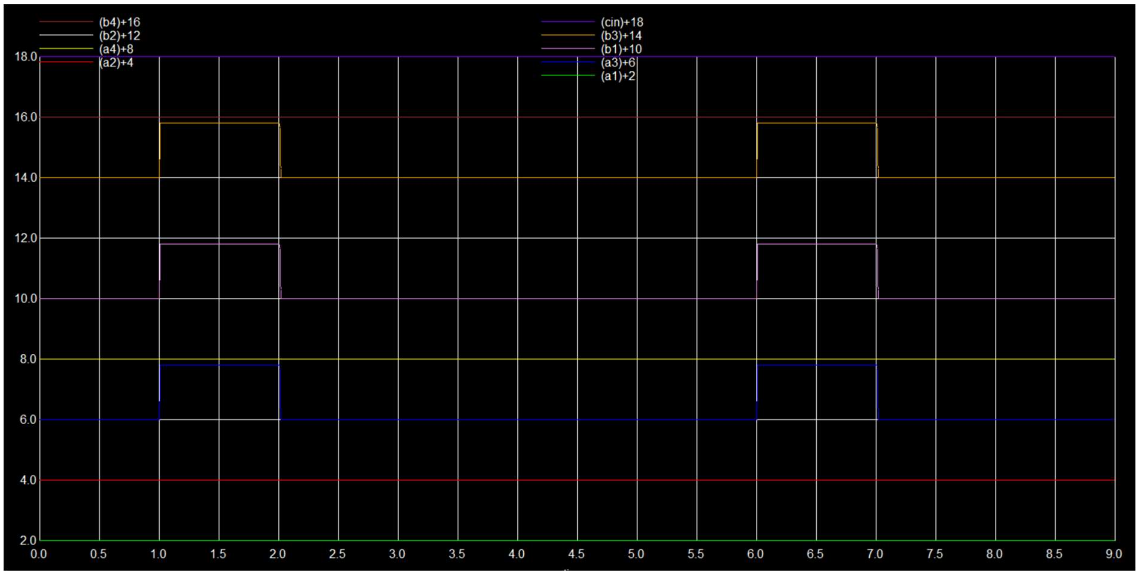
Script inputs:

```
VDD vdd 0 dc 1.8



Vin_1 A1 0 dc 0
Vin_5 B1 0 dc pulse(0 1.8 1000p 9p 9p 1n 5n)

Vin_2 A2 0 dc 0
Vin_6 B2 0 dc 0

Vin_3 A3 0 dc pulse(0 1.8 1000p 9p 9p 1n 5n)
Vin_7 B3 0 dc pulse(0 1.8 1000p 9p 9p 1n 5n)

Vin_4 A4 0 dc 0
Vin_8 B4 0 dc 0

Vin_9 cin 0 dc 0
```

Output:



Input plot:

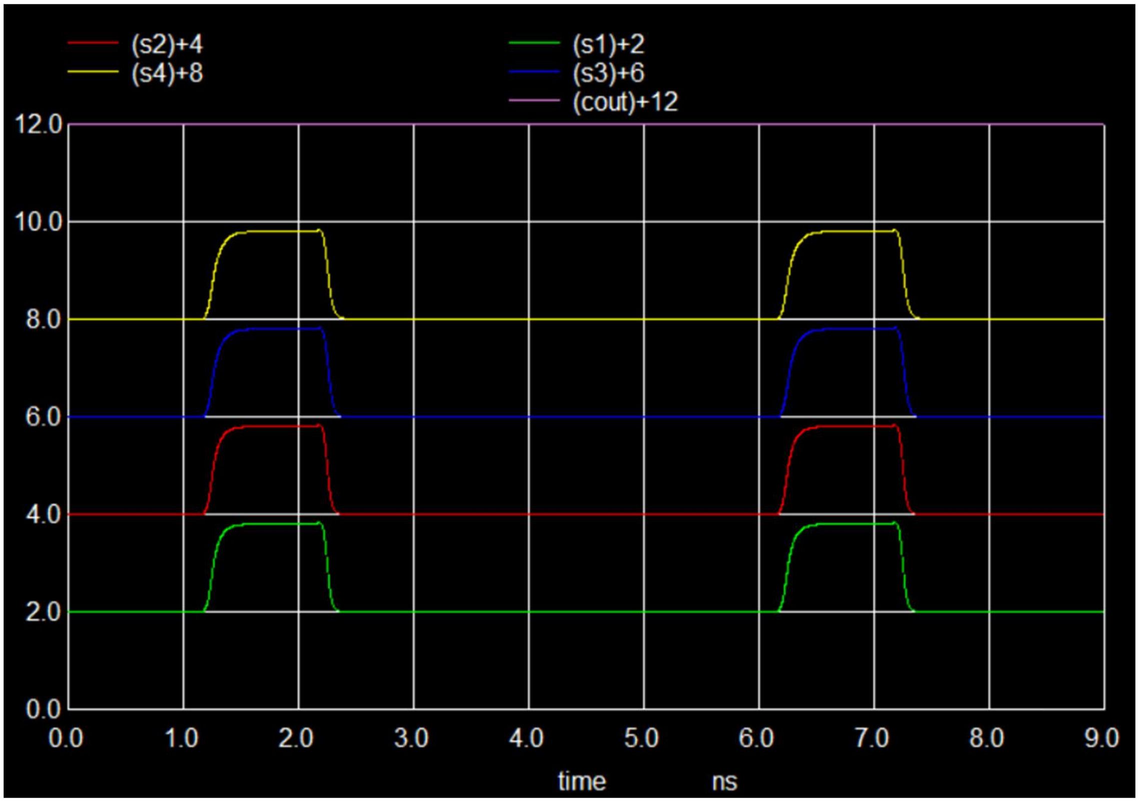


b)4-5: 0100+1011=1111

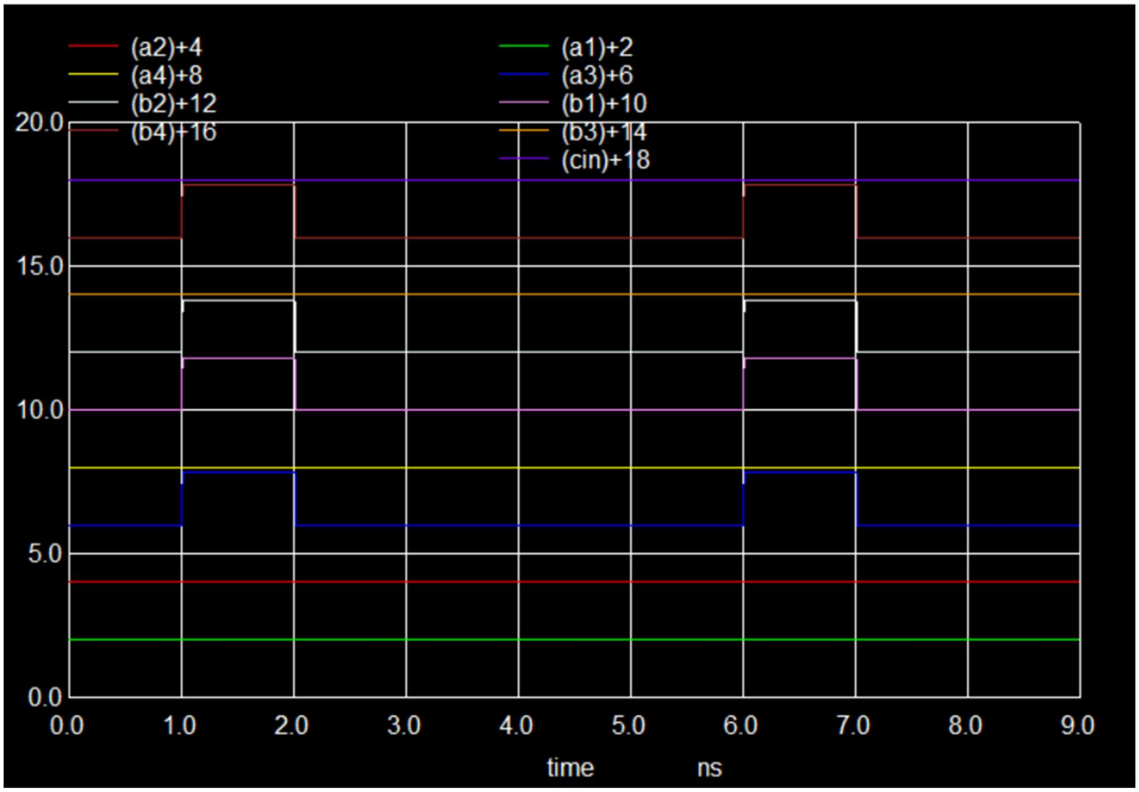
Script:

```
VDD vdd 0 dc 1.8

Vin_1 A1 0 dc pulse(0 0 1000p 9p 9p 1n 5n)
Vin_5 B1 0 dc pulse(0 1.8 1000p 9p 9p 1n 5n)

Vin_2 A2 0 dc pulse(0 0 1000p 9p 9p 1n 5n)
Vin_6 B2 0 dc pulse(0 1.8 1000p 9p 9p 1n 5n)

Vin_3 A3 0 dc pulse(0 1.8 1000p 9p 9p 1n 5n)
Vin_7 B3 0 dc pulse(0 0 1000p 9p 9p 1n 5n)

Vin_4 A4 0 dc pulse(0 0 1000p 9p 9p 1n 5n)
Vin_8 B4 0 dc pulse(0 1.8 1000p 9p 9p 1n 5n)

Vin_9 cin 0 dc pulse(0 0 1000p 9p 9p 1n 5n)
```

Output:


(s2)+4
(s4)+8
(s1)+2
(s3)+6
(cout)+12
12.0
10.0
8.0
6.0
4.0
2.0
0.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
time
ns

Input:


(a2)+4
(a4)+8
(b2)+12
(b4)+16
(a1)+2
(a3)+6
(b1)+10
(b3)+14
(cin)+18
20.0
15.0
10.0
5.0
0.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
time
ns

Ripple Delay Observation:

Script inputs:

VDD vdd 0 dc 1.8

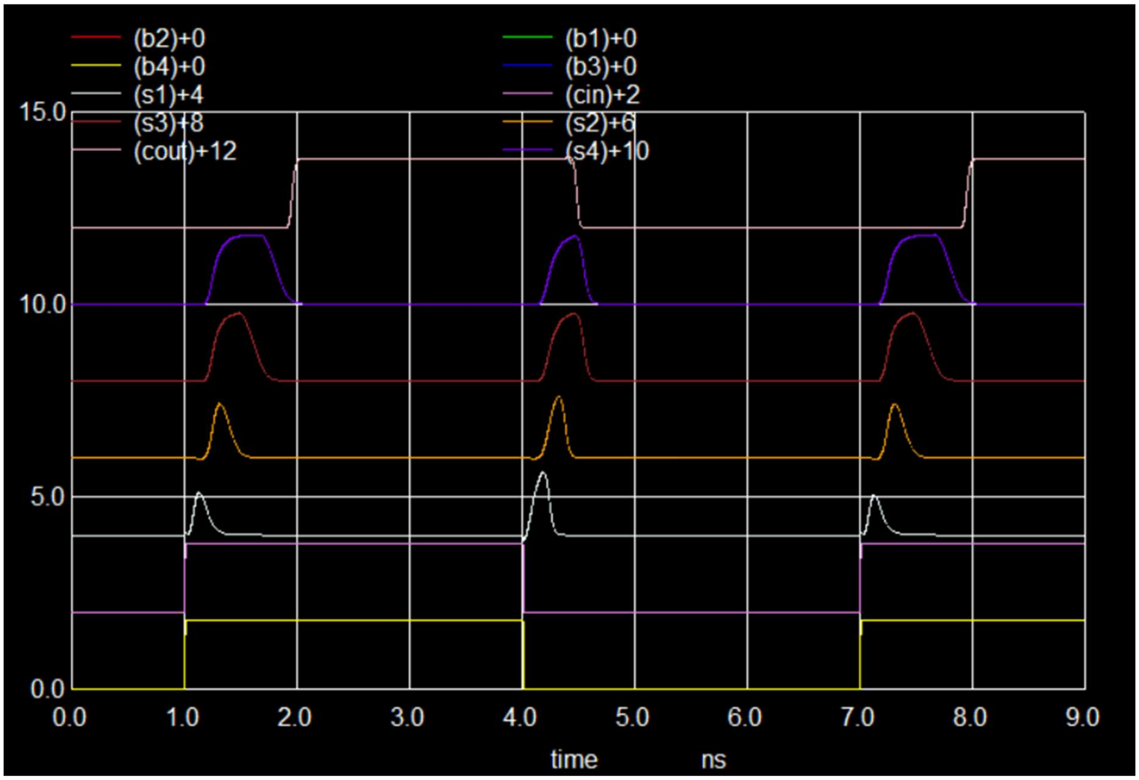
Vin_1 A1 0 dc pulse(0 0 1000p 9p 9p 3n 6n)
Vin_5 B1 0 dc pulse(0 1.8 1000p 9p 9p 3n 6n)

Vin_2 A2 0 dc pulse(0 0 1000p 9p 9p 3n 6n)
Vin_6 B2 0 dc pulse(0 1.8 1000p 9p 9p 3n 6n)

Vin_3 A3 0 dc pulse(0 0 1000p 9p 9p 3n 6n)
Vin_7 B3 0 dc pulse(0 1.8 1000p 9p 9p 3n 6n)

Vin_4 A4 0 dc pulse(0 0 1000p 9p 9p 3n 6n)
Vin_8 B4 0 dc pulse(0 1.8 1000p 9p 9p 3n 6n)

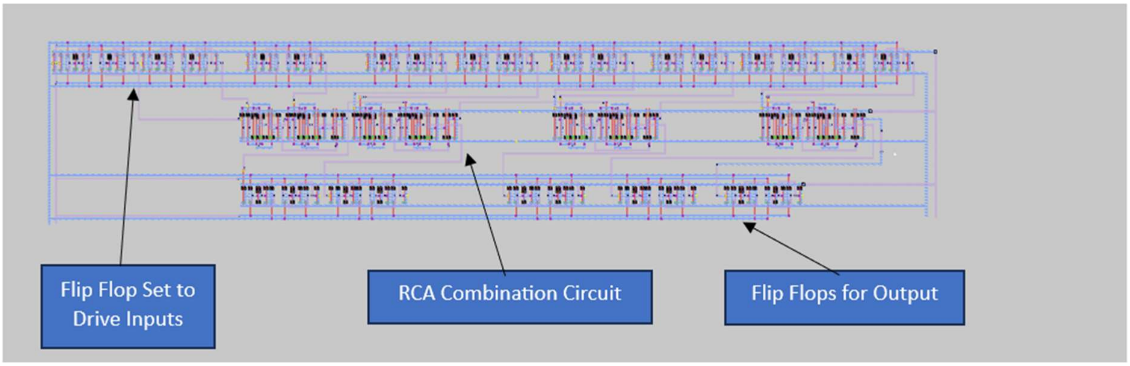
Vin_9 cin 0 dc pulse(0 1.8 1000p 9p 9p 3n 6n)



The rise and fall at initial switching sequence is less at S1 and more at s4, as it takes more time for S4 to calculate delay due to the ripple effect.

3.Use hierarchical design flow to add flip-flops to create a clocked version of this 4-bit RCA with inputs A[4:1] and B[4:1] and carry-input C$_{in}$ as shown on the next slide. Make sure it passes all DRCs, extract the circuit level netlist, and find the maximum clock frequency using SPICE. (30 points).

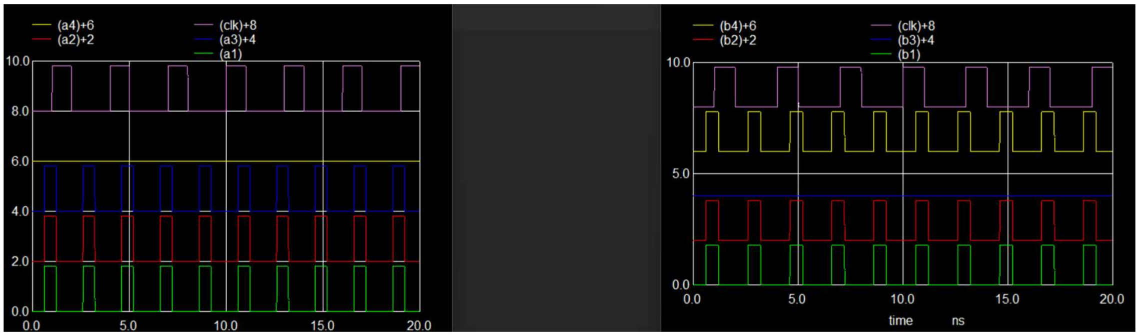
Magic Design:



Test Results:

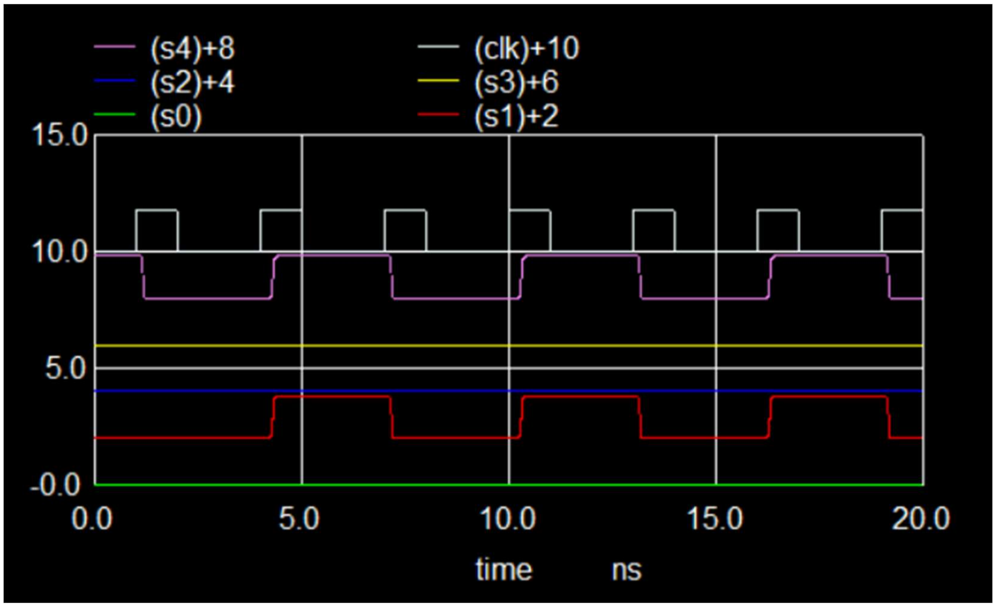
Test 1: Input: A4 A3 A2 A1 : 0111, B4 B3 B2 B1: 1011, CIN=0,

Outputs: S3 S2 S1 S0: 0010, Carry Out (S4)=1



Output:

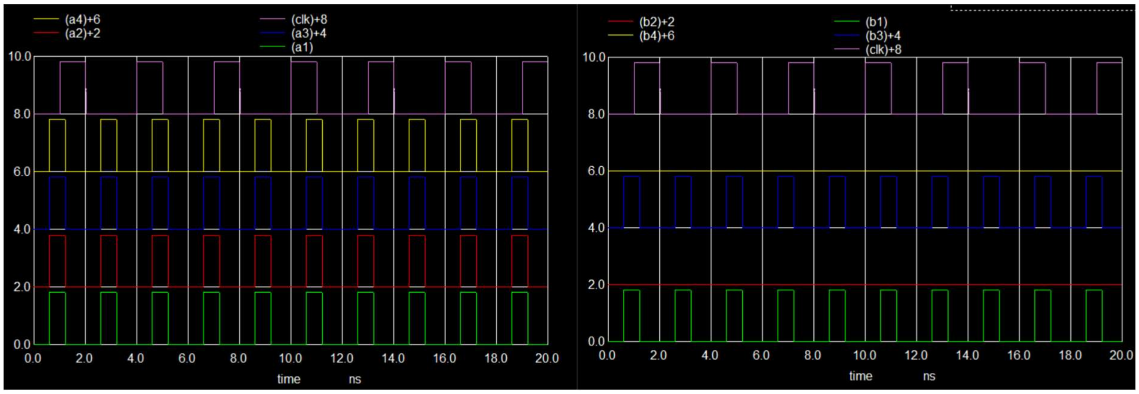
S3 S2 S1 S0: 0010, Carry Out (S4)=1



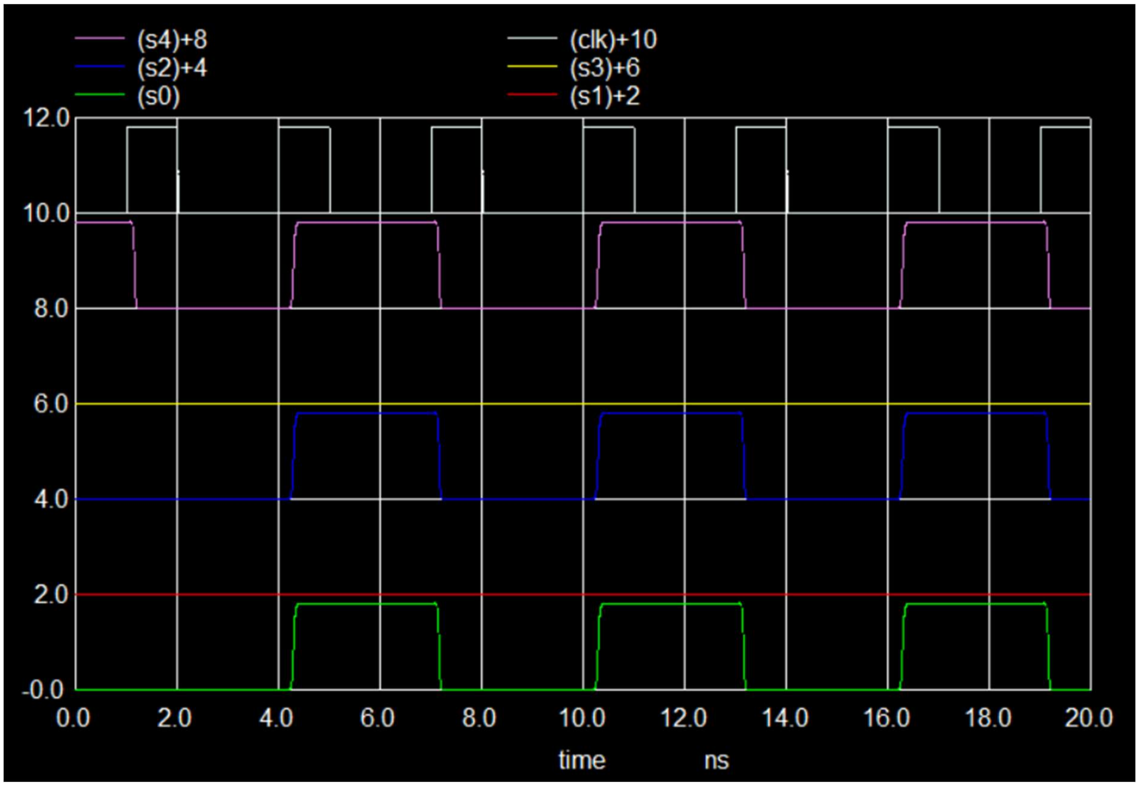
Test 2:

Input: A4 A3 A2 A1 : 1111, B4 B3 B2 B1: 0101, CIN=1,

Outputs: S3 S2 S1 S0: 0101, Carry Out (S4)=1



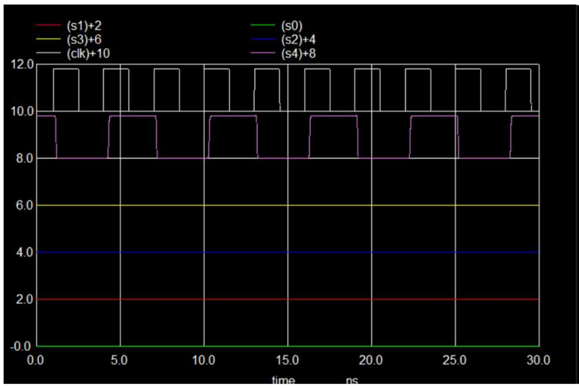
Output:

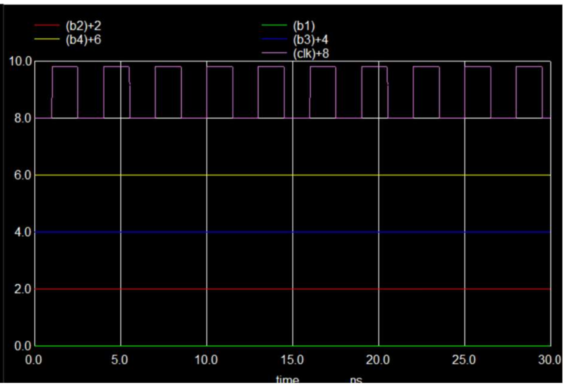


Frequency Check:

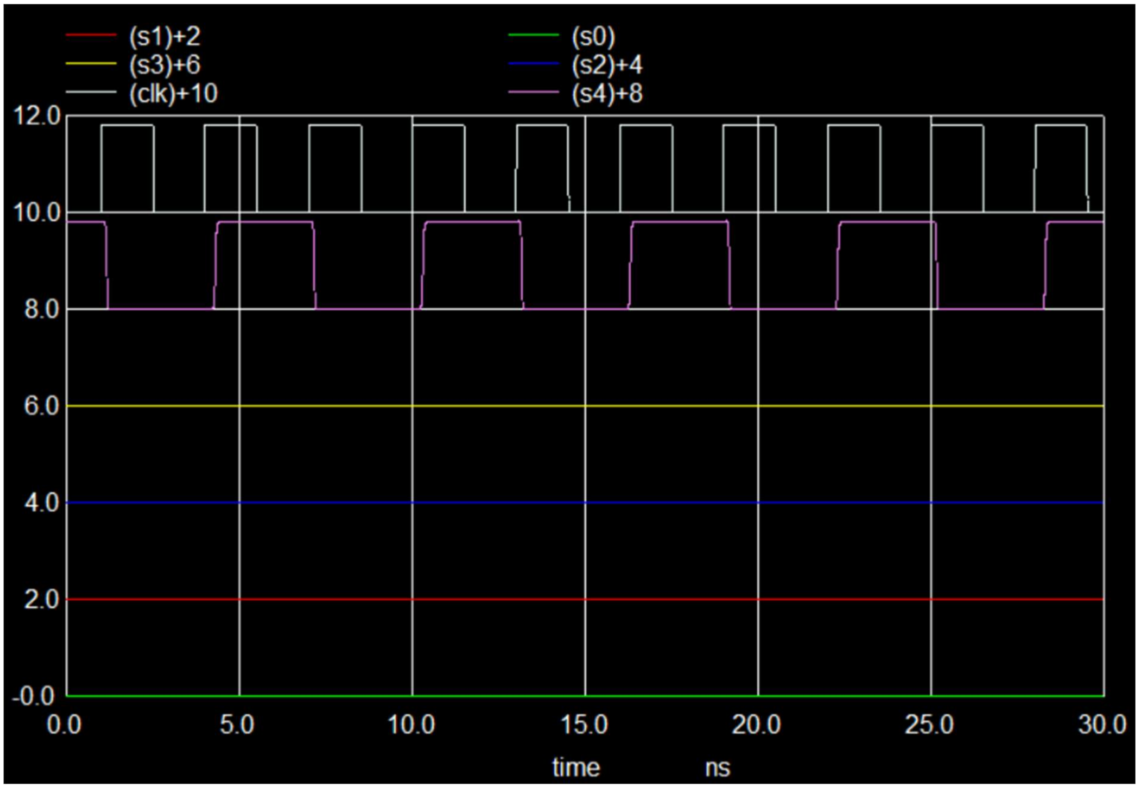
Critical Path: Input: A4 A3 A2 A1 : 1111, B4 B3 B2 B1: 0000, CIN=1

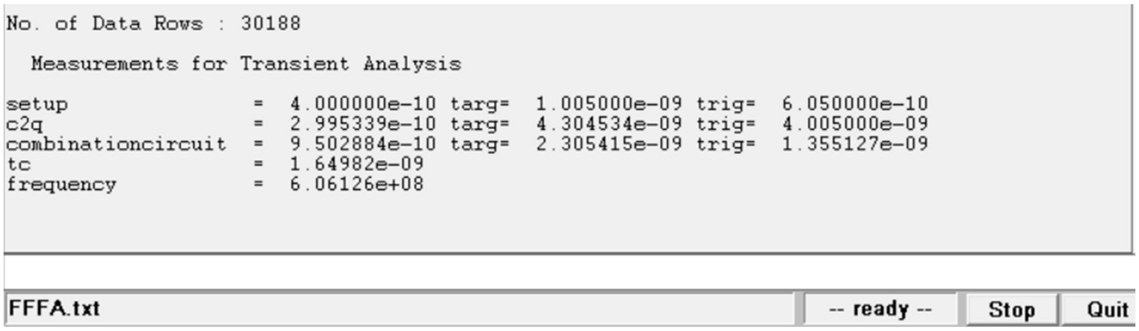
Output: S3 S2 S1 S0: 0000, Carry Out (S4)=1





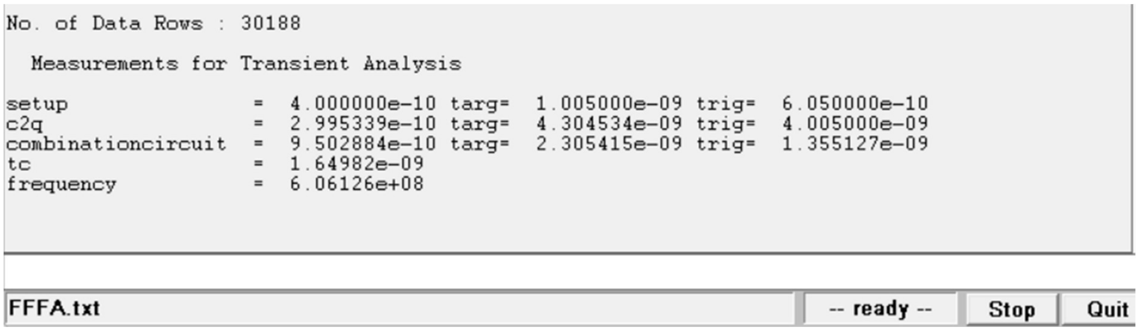
Output:





```
No. of Data Rows : 30188

  Measurements for Transient Analysis

setup                = 4.000000e-10 targ= 1.005000e-09 trig= 6.050000e-10
c2q                  = 2.995339e-10 targ= 4.304534e-09 trig= 4.005000e-09
combinationcircuit   = 9.502884e-10 targ= 2.305415e-09 trig= 1.355127e-09
tc                   = 1.64982e-09
frequency            = 6.06126e+08
```

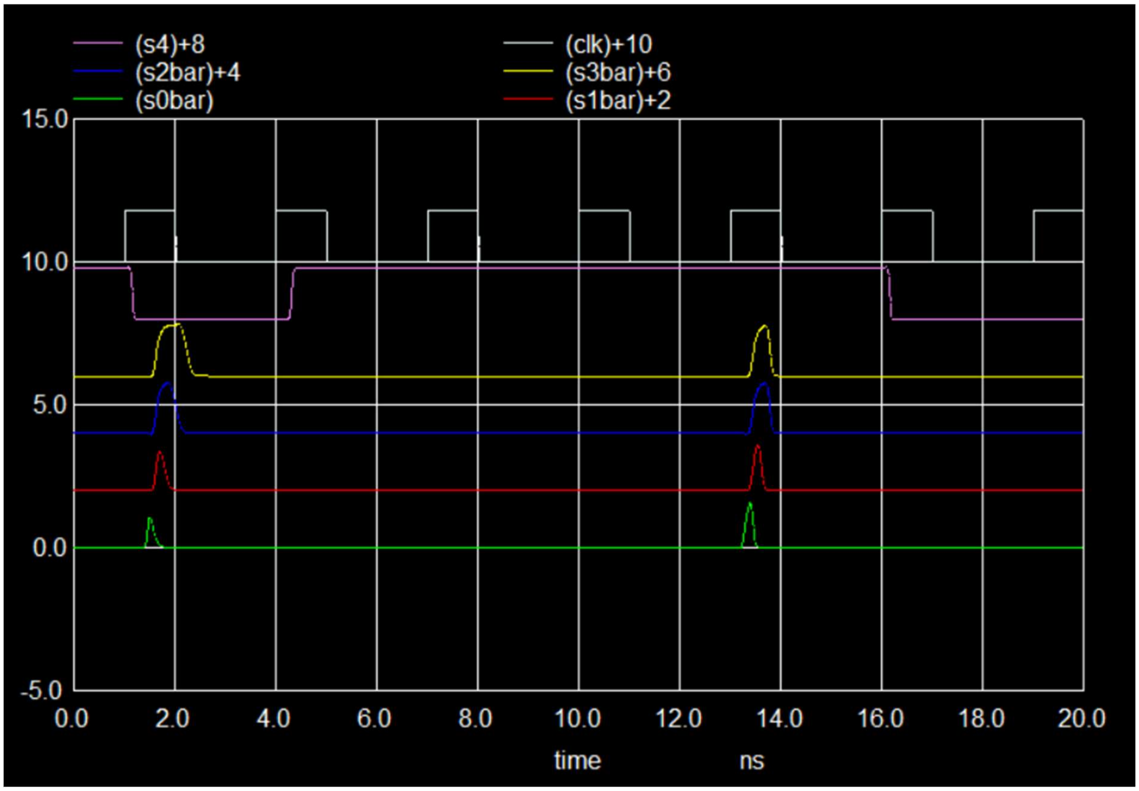
Combination Path: Ciin to Cout (S4bar) of full adder.



```
No. of Data Rows : 30188

  Measurements for Transient Analysis

setup                = 4.000000e-10 targ= 1.005000e-09 trig= 6.050000e-10
c2q                  = 2.995339e-10 targ= 4.304534e-09 trig= 4.005000e-09
combinationcircuit   = 9.502884e-10 targ= 2.305415e-09 trig= 1.355127e-09
tc                   = 1.64982e-09
frequency            = 6.06126e+08
```

Max Frequency received: 606 Mhz

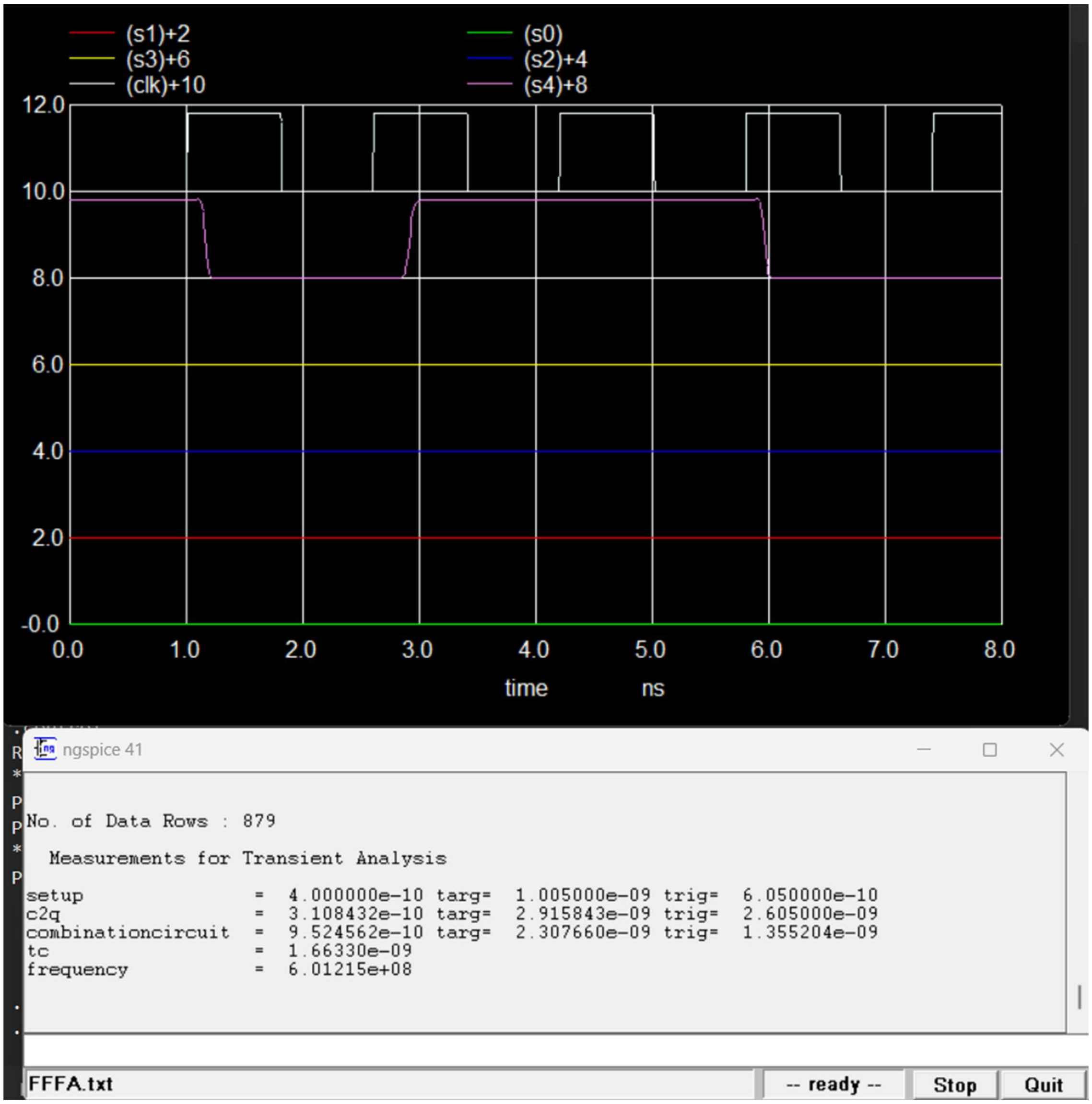
Other Observation, Ripple Time variation: The outputs of FA for each sum icrease with increase in MSB bit, this delay is due to the ripple effect.



Result when trying to run with this frequency: 1/Max Frequency =1/606 Mhz=1600ps

Clock in spice file: V0 clk 0 dc pulse(0 1.8 1000p 10p 10p 800p 1600p)

Output:



```
No. of Data Rows : 879

  Measurements for Transient Analysis

setup                  =  4.000000e-10 targ=  1.005000e-09 trig=  6.050000e-10
c2q                    =  3.108432e-10 targ=  2.915843e-09 trig=  2.605000e-09
combinationcircuit     =  9.524562e-10 targ=  2.307660e-09 trig=  1.355204e-09
tc                     =  1.66330e-09
frequency              =  6.01215e+08
```

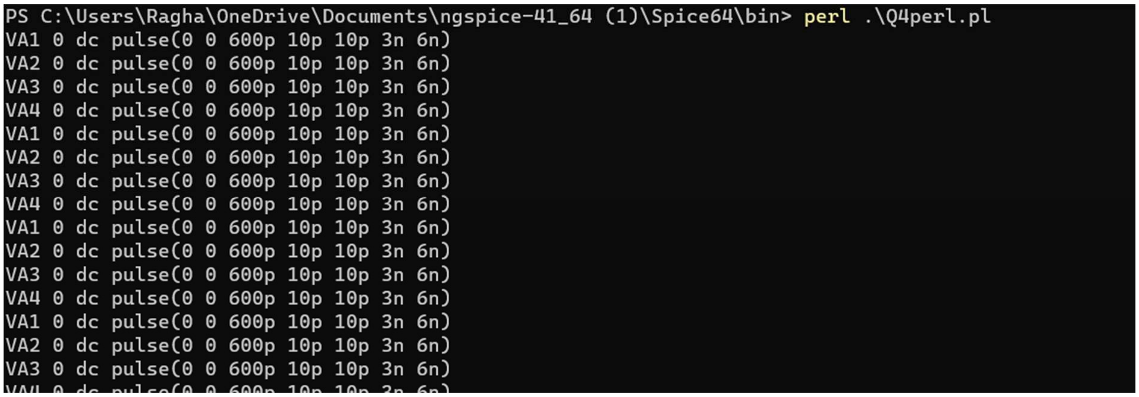
4. **Extra credit question: Write a Perl script to automatically tabulate the average power consumed for addition operation for all possible input combinations of this clocked 4-bit RCA i.e. 2^9 = 512 possibilities. (10 points).**

I was able to write a script like this:

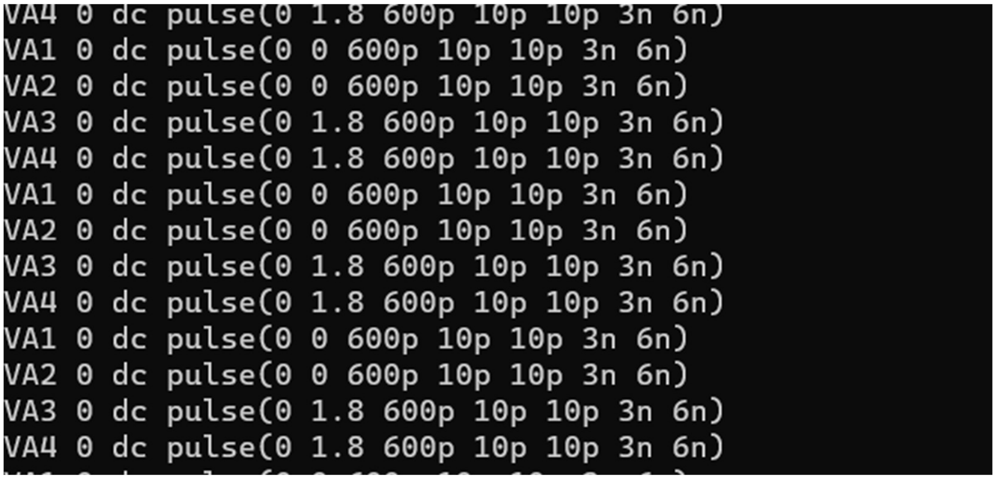
```
#!/usr/bin/perl
use strict;
use warnings;

my $filename = 'Q4FinalExam.txt';
```

```
# Generate all possible binary combinations for 8 inputs (A1 to B4)
for (my $i = 0; $i <= 255; $i++) {
    open(my $fh, '<', $filename) or die "Could not open file '$filename' $!";

    my $binary = sprintf("%08b", $i);
    my @binary_array = split //, $binary;

    while (my $line = <$fh>) {
        chomp($line);
        if ($line =~ /^V\d+\s+(A\d+)\s+0\s+dc\s+pulse\((.*?)\)$/) {
            my $input_name = $1;
            my $original_params = $2;

            my @params = split /\s+/, $original_params;

            # Ensure @binary_array has elements before using shift
            if (@binary_array) {
                $params[1] = shift(@binary_array) * 1.8; # Replace the second
parameter with the corresponding bit value
            } else {
                warn "Not enough bits in binary_array for line: $line\n";
            }

            my $new_line = "V$input_name 0 dc pulse(" . join(' ', @params) .
")";
            print "$new_line\n";
        }
    }

    close $fh;
}

# Execute ngspice
my $ngspice_command = "ngspice -b $filename";
`$ngspice_command`;
```

The execution is yielding



```
PS C:\Users\Ragha\OneDrive\Documents\ngspice-41_64 (1)\Spice64\bin> perl .\Q4perl.pl
VA1 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 0 600p 10p 10p 3n 6n)
```



```
VA4 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
```

However, the script is unable to invoke ngspice and write the power to power.txt file

5. **Repeat the simulation part in question 3 at half the supply voltage i.e. VDD = 0.9V and find the new**
maximum clock frequency. Repeat question 4 at this new value

Maximum Frequency calculation in Spice:

VDD vdd 0 dc 0.9

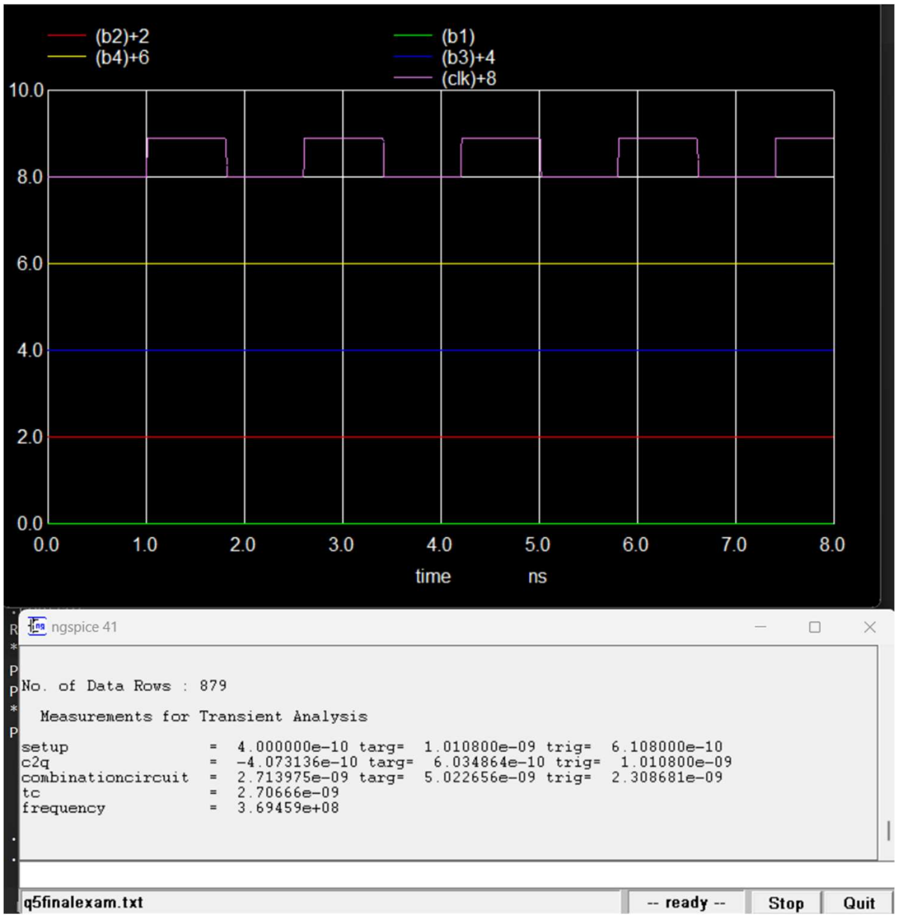
V0 clk 0 dc pulse(0 0.9 1000p 10p 10p 800p 1600p)

V1 A1 0 dc pulse(0 0.9 600p 10p 10p 3n 6n)
V2 A2 0 dc pulse(0 0.9 600p 10p 10p 3n 6n)
V3 A3 0 dc pulse(0 0.9 600p 10p 10p 3n 6n)
V4 A4 0 dc pulse(0 0.9 600p 10p 10p 3n 6n)

V5 B1 0 dc pulse(0 0 600p 10p 10p 3n 6n)
V6 B2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
V7 B3 0 dc pulse(0 0 600p 10p 10p 3n 6n)

V8 B4 0 dc pulse(0 0 600p 10p 10p 3n 6n)

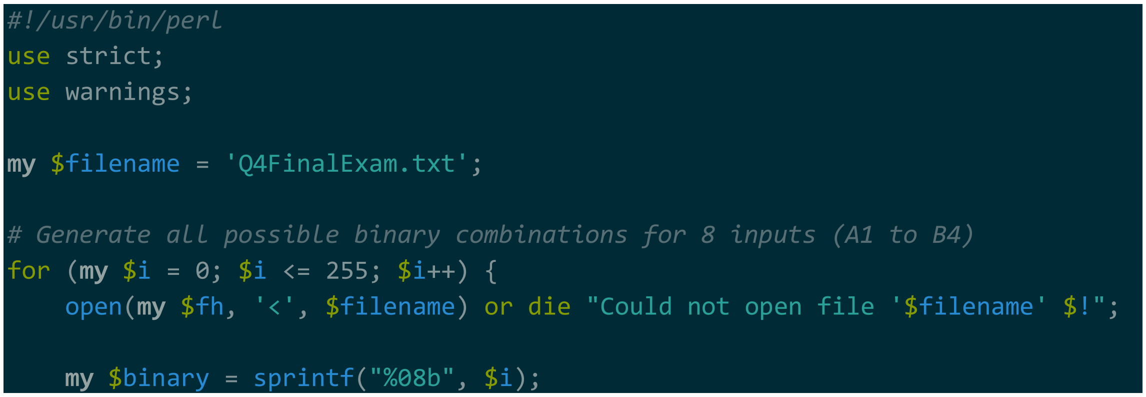
V9 Cin 0 dc pulse(0 0.9 600p 10p 10p 3n 6n)



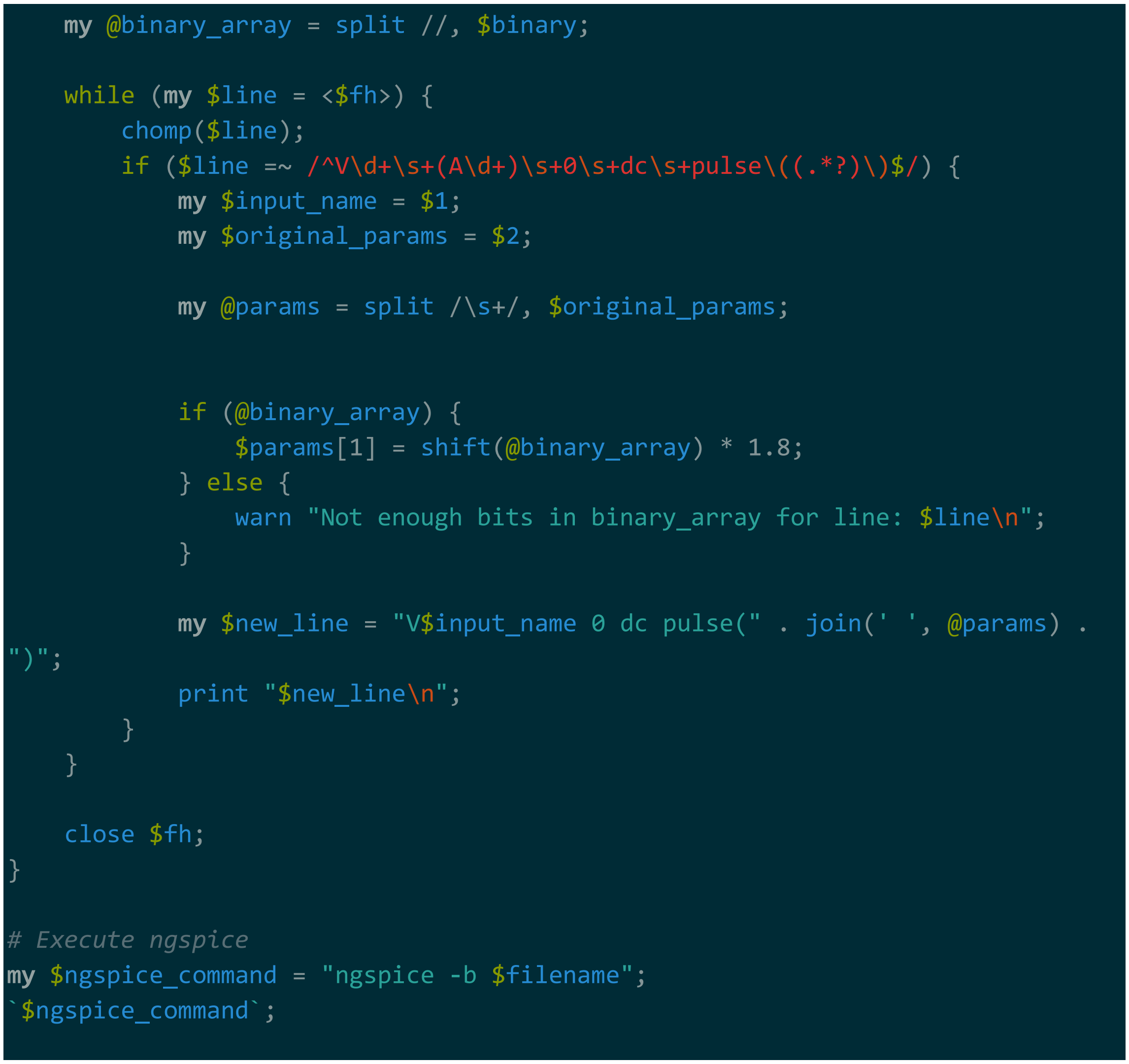
```
No. of Data Rows : 879

  Measurements for Transient Analysis

setup                 =  4.000000e-10 targ=  1.010800e-09 trig=  6.108000e-10
c2q                   =  -4.073136e-10 targ=  6.034864e-10 trig=  1.010800e-09
combinationcircuit    =  2.713975e-09 targ=  5.022656e-09 trig=  2.308681e-09
tc                    =  2.70666e-09
frequency             =  3.69459e+08
```

Frequency decreased from 606Mhz to 369Mhz after decreasing the voltage by half.

For perl script simulation, code written:

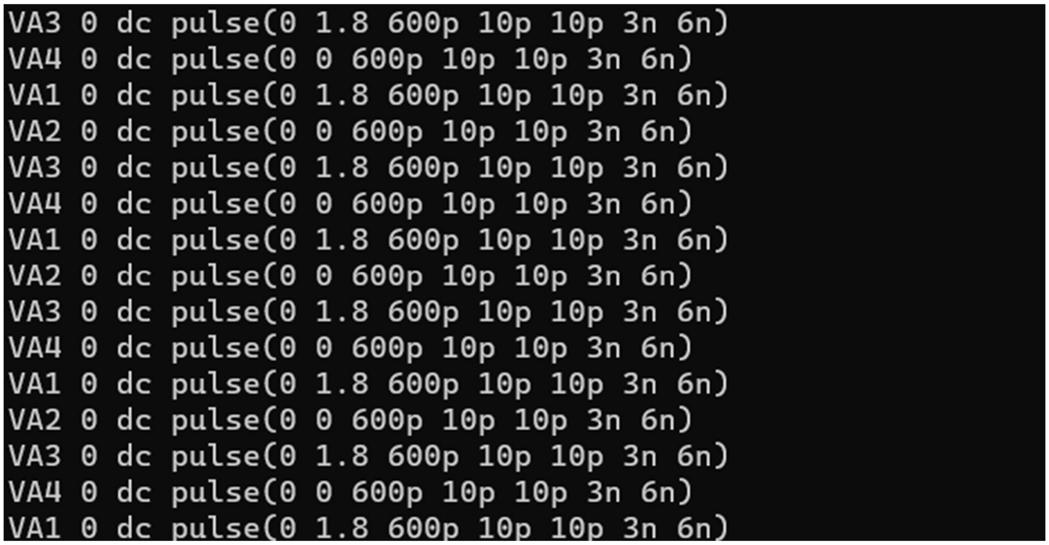


```
#!/usr/bin/perl
use strict;
use warnings;

my $filename = 'Q4FinalExam.txt';

# Generate all possible binary combinations for 8 inputs (A1 to B4)
for (my $i = 0; $i <= 255; $i++) {
    open(my $fh, '<', $filename) or die "Could not open file '$filename' $!";

    my $binary = sprintf("%08b", $i);
```



```
    my @binary_array = split //, $binary;

    while (my $line = <$fh>) {
        chomp($line);
        if ($line =~ /^V\d+\s+(A\d+)\s+0\s+dc\s+pulse\((.*?)\)$/) {
            my $input_name = $1;
            my $original_params = $2;

            my @params = split /\s+/, $original_params;


            if (@binary_array) {
                $params[1] = shift(@binary_array) * 1.8;
            } else {
                warn "Not enough bits in binary_array for line: $line\n";
            }

            my $new_line = "V$input_name 0 dc pulse(" . join(' ', @params) .
")";
            print "$new_line\n";
        }
    }

    close $fh;
}

# Execute ngspice
my $ngspice_command = "ngspice -b $filename";
`$ngspice_command`;
```

Code not working as expected. Actual result is:



```
VA3 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA2 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA3 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
VA4 0 dc pulse(0 0 600p 10p 10p 3n 6n)
VA1 0 dc pulse(0 1.8 600p 10p 10p 3n 6n)
```

It is able to create different input combinations but could not invoke and execute ngspice