Test Case 1:

Assembly Instructions:

```
load r5, 0(r2)    # r5 = 0
load r2, 1(r2)    # r2 = 0
div r0, r2, r4    # r0 = 0
add r4, r2, r6    # r4 = 5 + 6 = 11
bne r1, r2, 2     # 1 != 5 -> true so branch taken
add r2, r1, r5
ret               # (after call) jumps to last add
inst
add r6, r1, r5    # r6 = 1 + 0 = 1
bne r6, r6, 5     # 1 != 1 -> false so branch not
taken
store r1, 2(r2)   # mem[7] = 1
bne r1, r2, 0     # 1 != 5 -> true so branch taken
call 6            # jumps to address 6
add r0, r0 , r0   # r0 = r0
```

Initial Register Values:

```
Reg = {
   "r0": 0,
   "r1": 1,
   "r2": 2,
   "r3": 3,
   "r4": 4,
   "r5": 5,
   "r6": 6,
   "r7": 7,
   "r8": 8
}
```

Memory: `Mem[0] = 2, Mem[1] = 5.`
Starting address: `0`
Results:

---------- Tomasulo Algorithm Simulator ----------

Clock Cycle: 28

Trace Table:

instruction	issue	start_exec	end_exec	write
['load', 'r5', '0', 'r2']	1	2	4	5
['load', 'r2', '1', 'r2']	2	3	5	6
['div', 'r0', 'r2', 'r4']	3	7	16	17
['add', 'r4', 'r2', 'r6']	4	7	8	9
['bne', 'r1', 'r2', '2']	5	7	7	8
['add', 'r6', 'r1', 'r5']	9	10	11	12
['bne', 'r6', 'r6', '5']	10	13	13	14
['store', 'r1', 2, 'r2']	11	15	17	19
['bne', 'r1', 'r2', 0]	15	16	16	18
['call', 6, None, None]	19	20	20	21
['ret', None, None, None]	22	23	23	24
['add', 'r0', 'r0', 'r0']	25	26	27	28

Resrvation Station:

Name	Busy	OP	Vj	Vk	Qj	Qk	A	Imm
Load1	N							
Load2	N							
Store1	N							
Store2	N							
Add/Addi_1	N							
Add/Addi_2	N							
Add/Addi_3	N							
Nand	N							
Div	N							
Bne	N							
Call/Ret	N							

Register Stat Table:

r0	r1	r2	r3	r4	r5	r6	r7	r8

Register Values:

r0	r1	r2	r3	r4	r5	r6	r7	r8
0	12	0	3	6	0	1	7	8

Press Enter to continue or 0 to Exit:
0

Number of Branches: 3
Number of Branches Taken: 2
Branch Misprediction: 0.6666666666666666 %

IC: 12
Clock Cycles: 28
IPC: 0.42857142857142855

Test Case 2:

Assembly Instructions:

```
addi r1, r1, 1        # r1 = 8
addi r2, r2, 8        # r2 = 8
nand r3, r1, r2       # r3 = -9
div r4, r1, r2        # r4 = 1
store r3, 0(r0)       # Mem[0] = -9
add r0, r3, r4        # r0 = 0
```

Initial Register Values:

```
Reg = {
   "r0": 0,
   "r1": 1,
   "r2": 2,
   "r3": 3,
   "r4": 4,
   "r5": 5,
   "r6": 6,
   "r7": 7,
   "r8": 8
}
```

Memory: `Initialized with 0s`
Starting address: `0`
Results:

---------- Tomasulo Algorithm Simulator ----------

Clock Cycle: 19

Trace Table:

instruction	issue	start_exec	end_exec	write
['addi', 'r1', 'r1', '1']	1	2	3	4
['addi', 'r2', 'r2', '8']	2	3	4	5
['nand', 'r3', 'r1', 'r2']	3	6	6	7
['div', 'r4', 'r1', 'r2']	4	6	15	16
['store', 'r3', '0', 'r0']	5	8	10	11
['add', 'r0', 'r3', 'r4']	6	17	18	19

Resrvation Station:

Name	Busy	OP	Vj	Vk	Qj	Qk	A	Imm
Load1	N							
Load2	N							
Store1	N							
Store2	N							
Add/Addi_1	N							
Add/Addi_2	N							
Add/Addi_3	N							
Nand	N							
Div	N							
Bne	N							
Call/Ret	N							

Register Stat Table:

r0	r1	r2	r3	r4	r5	r6	r7	r8

Register Values:

r0	r1	r2	r3	r4	r5	r6	r7	r8
0	2	10	-3	0.2	5	6	7	8

Press Enter to continue or 0 to Exit:
0

Number of Branches: 0
Number of Branches Taken: 0
Branch Misprediction: 0%

IC: 6
Clock Cycles: 19
IPC: 0.3157894736842105

Test Case 3:

Assembly Instructions:

```
div r2, r6, r1  #r2 = 6
add r3, r2, r2  #r3 = 12
nand r2, r6, r1       #r2 = -1
```

Initial Register Values:

```
Reg = {
   "r0": 0,
   "r1": 1,
   "r2": 2,
   "r3": 3,
   "r4": 4,
   "r5": 5,
   "r6": 6,
   "r7": 7,
   "r8": 8
}
```

Memory: Initialized with 0s
Starting Address: 0
Results:

---------- Tomasulo Algorithm Simulator ----------

Clock Cycle: 15

Trace Table:

instruction	issue	start_exec	end_exec	write
['div', 'r2', 'r6', 'r1']	1	2	11	12
['add', 'r3', 'r2', 'r2']	2	13	14	15
['nand', 'r2', 'r6', 'r1']	3	4	4	5

Resrvation Station:

Name	Busy	OP	Vj	Vk	Qj	Qk	A	Imm
Load1	N							
Load2	N							
Store1	N							
Store2	N							
Add/Addi_1	N							
Add/Addi_2	N							
Add/Addi_3	N							
Nand	N							
Div	N							
Bne	N							
Call/Ret	N							

Register Stat Table:

r0	r1	r2	r3	r4	r5	r6	r7	r8

Register Values:

r0	r1	r2	r3	r4	r5	r6	r7	r8
0	1	-1	12	4	5	6	7	8

Press Enter to continue or 0 to Exit:
0

Number of Branches: 0
Number of Branches Taken: 0
Branch Misprediction: 0%

IC: 3
Clock Cycles: 15
IPC: 0.2