```
.include "/nominal.jsim"
.include "/stdcell.jsim"
.include "/2dcheckoff_3ns.jsim"

.subckt getPG A B P G
Xpinv A B Pinv nor2
Xp Pinv P inverter
Xginv A B Ginv nand2
Xg Ginv G inverter
.ends

.subckt getnextPG g1 p1 g2 p2 g3 p3
Xp3ing p1 p2 p3inv nand2
Xp3 p3inv p3 inverter
Xpg g1 p2 pg nand2
Xg2inv g2 g2inv inverter
Xg3 pg g2inv g3 nand2
.ends


.subckt look_ahead_carry P G Cin Cout
Xpcinv P Cin PCinv nand2
Xginv G Ginv inverter
Xc PCinv Ginv Cout nand2
.ends

.subckt look_ahead_sum A B C S
Xxorab A B P xor2
Xxorpc P C S xor2
.ends

//Carry look-ahead adder
.subckt heirarch_adder A[15:0] B[15:0] C[0] S[15:0] Cout
Xpg A[15:0] B[15:0] P[15:0] G[15:0] getPG

Xlvl_1 G[0:14:2] P[0:14:2] G[1:15:2] P[1:15:2] Gi[1:15:2] Pi[1:15:2] getnextPG
Xlvl_2 Gi[1:15:4] Pi[1:15:4] Gi[3:15:4] Pi[3:15:4] Gj[3:15:4] Pj[3:15:4] getnextPG
Xlvl_3 Gj[3:15:8] Pj[3:15:8] Gj[7:15:8] Pj[7:15:8] Gk[7:15:8] Pk[7:15:8] getnextPG

Xcarry_lvl1 Pi[1] Gi[1] C[0] C[2] look_ahead_carry
Xcarry_lvl2 Pj[3] Gj[3] C[0] C[4] look_ahead_carry
Xcarry_lvl3 Pk[7] Gk[7] C[0] C[8] look_ahead_carry
Xcarry_even Pi[5:14:4] Gi[5:14:4] C[4:12:4] C[6:14:4] look_ahead_carry
Xcarry12 Pj[11] Gj[11] C[8] C[12] look_ahead_carry
Xcarryodd P[14:0:2] G[14:0:2] C[14:0:2] C[15:0:2] look_ahead_carry
Xcarry16 Pk[15] Gk[15] C[8] Cout look_ahead_carry

Xsum A[15:0] B[15:0] C[15:0] S[15:0] look_ahead_sum
.ends

//Full adder
.subckt adder32 op0 A[31:0] B[31:0] S[31:0] Z V N
Xbinv op0#32 B[31:0] XB[31:0] xor2
//Xbufferop op0 op_buff buffer_8
//XselB op_buff#32 B[31:0] Binv[31:0] XB[31:0] mux2

Xadd1 A[15:0] XB[15:0] op0 S[15:0] C16 heirarch_adder

//Carry-select adder
X1 1 constant1
Xadd2 A[31:16] XB[31:16] 0 S1[31:16] C1_32 heirarch_adder
Xadd3 A[31:16] XB[31:16] 1 S2[31:16] C2_32 heirarch_adder
Xsel C16#16 S1[31:16] S2[31:16] S[31:16] mux2



//Find Z
Xnor4 S[7:0] S[15:8] S[23:16] S[31:24] nor_out[7:0] nor4
Xand4 nor_out[1:0] nor_out[3:2] nor_out[5:4] nor_out[7:6] nand_out[1:0] nand4
Xnor2 nand_out[1] nand_out[0] Z nor2



//Find V
Xinv1 A[31] Ainv inverter
Xinv2 XB[31] XBinv inverter
Xinv3 S[31] Sinv inverter
Xand_1 A[31] XB[31] Sinv out1 nand3
Xand_2 Ainv XBinv S[31] out2 nand3
Xorv out1 out2 V nand2

//Find N
.connect S[31] N
.ends
```