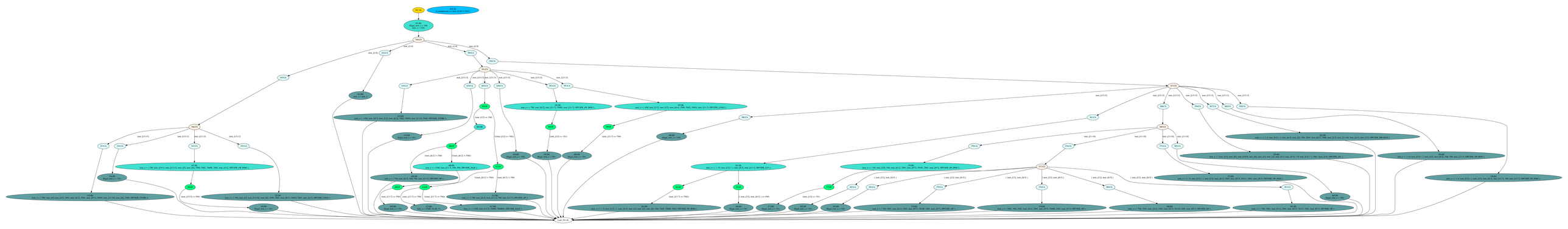

383:BS
illegal_instr_o = 1'bX;

Leaf_341:AL

369:CS

370:CA

instr_i[11:10]

376:CA

instr_i[11:10]

375:CA

instr_i[11:10]

385:CA

instr_i[11:10]

370:BL

instr_o = { 1'b0, instr_i[10], 5'b0, instr_i[6:2], 2'b01, instr_i[9:7], 3'b101, 2'b01, instr_i[9:7], OPCODE_OP_IMM }; 377:CS

375:BS
instr_o = { { 6{ instr_i[12] } }, instr_i[12], instr_i[6:2], 2'b01, instr_i[9:7], 3'b111, 2'b01, instr_i[9:7], OPCODE_OP_IMM };

385:BS
illegal_instr_o = 1'bX;

388:CA

388:BS
illegal_instr_o = 1'bX;

353:BS
instr_o = { 5'b0, instr_i[5], instr_i[12], 2'b01, instr_i[4:2], 2'b01, instr_i[9:7], 3'b010, instr_i[11:10], instr_i[6], 2'b00, OPCODE_STORE };

399:IF

400:BS

illegal_instr_o = 1'b1;

(instr_i[11:7] == 5'b0)

341:AL

341:BL

illegal_instr_o = 1'b0;
instr_o = 1'bX;

344:CS

418:CA

418:BS
instr_o = { 4'b0, instr_i[8:7], instr_i[12], instr_i[6:2], 5'h02, 3'b010, instr_i[11:9], 2'b00, OPCODE_STORE };

354:BS
illegal_instr_o = 1'b1;

404:IF

405:BS
instr_o = { 7'b0, instr_i[6:2], 5'b0, 3'b0, instr_i[11:7], OPCODE_OP };

(instr_i[6:2] != 5'b0)

406:BL
instr_o = { 12'b0, instr_i[11:7], 3'b0, 5'b0, OPCODE_JALR };

!((instr_i[6:2] != 5'b0))

408:IF

345:CA

instr_i[1:0] 422:CA

instr_i[1:0]

390:CA

instr_i[1:0]

356:CA

instr_i[1:0]

346:CS

422:BS
instr_o = instr_i;

391:CS

357:CS

420:BS

illegal_instr_o = 1'bX;

372:IF

373:BS
illegal_instr_o = 1'b1;

(instr_i[12] == 1'b1)

353:CA

instr_i[15:13]

354:CA

instr_i[15:13]

347:CA

instr_i[15:13]

352:CA

instr_i[15:13]

414:IF

415:BS
instr_o = 32'h00_10_00_73;

(instr_i[11:7] == 5'b0)

417:BS
instr_o = { 12'b0, instr_i[11:7], 3'b000, 5'b00001, OPCODE_JALR };

!((instr_i[11:7] == 5'b0))

361:CA

361:BL
instr_o = { { 15{ instr_i[12] } }, instr_i[6:2], instr_i[11:7], OPCODE_LUI };

363:IF 365:IF

instr_i[15:13] instr_i[15:13]

368:CA

instr_i[15:13]

359:CA

instr_i[15:13]

387:CA

instr_i[15:13]

360:CA

instr_i[15:13]

358:CA

instr_i[15:13]

359:BS
instr_o = { instr_i[12], instr_i[8], instr_i[10:9], instr_i[6], instr_i[7], instr_i[2], instr_i[11], instr_i[5:3], { 9{ instr_i[12] } }, 4'b0, ~instr_i[15], OPCODE_JAL };

387:BS
instr_o = { { 4{ instr_i[12] } }, instr_i[6:5], instr_i[2], 5'b0, 2'b01, instr_i[9:7], 2'b00, instr_i[13], instr_i[11:10], instr_i[4:3], instr_i[12], OPCODE_BRANCH };

360:BS
instr_o = { { 6{ instr_i[12] } }, instr_i[12], instr_i[6:2], 5'b0, 3'b0, instr_i[11:7], OPCODE_OP_IMM };

358:BS
instr_o = { { 6{ instr_i[12] } }, instr_i[12], instr_i[6:2], instr_i[11:7], 3'b0, instr_i[11:7], OPCODE_OP_IMM };

347:BL
instr_o = { 2'b0, instr_i[10:7], instr_i[12:11], instr_i[5], instr_i[6], 2'b00, 5'h02, 3'b000, 2'b01, instr_i[4:2], OPCODE_OP_IMM };

352:BS
instr_o = { 5'b0, instr_i[5], instr_i[12:10], instr_i[6], 2'b00, 2'b01, instr_i[9:7], 3'b010, 2'b01, instr_i[4:2], OPCODE_LOAD };

413:BS
instr_o = { 7'b0, instr_i[6:2], instr_i[11:7], 3'b0, instr_i[11:7], OPCODE_OP };

419:CA

419:BS
illegal_instr_o = 1'b1;

382:CA

382:BS
illegal_instr_o = 1'b1;

394:IF

395:BS
illegal_instr_o = 1'b1;

(instr_i[12] == 1'b1)

402:CA

403:IF

403:BL

(instr_i[12] == 1'b0)

412:IF

!((instr_i[12] == 1'b0))

381:BS
instr_o = { 7'b0, 2'b01, instr_i[4:2], 2'b01, instr_i[9:7], 3'b111, 2'b01, instr_i[9:7], OPCODE_OP };

425:AS
is_compressed_o = instr_i[1:0] != 2'b11;

350:BS
illegal_instr_o = 1'b1;

{ instr_i[12], instr_i[6:5] }

383:CA

{ instr_i[12], instr_i[6:5] }

379:CA

{ instr_i[12], instr_i[6:5] }

378:CA

{ instr_i[12], instr_i[6:5] }

380:CA

{ instr_i[12], instr_i[6:5] }

381:CA

{ instr_i[12], instr_i[6:5] }

379:BS
instr_o = { 7'b0, 2'b01, instr_i[4:2], 2'b01, instr_i[9:7], 3'b100, 2'b01, instr_i[9:7], OPCODE_OP };

378:BS
instr_o = { 2'b01, 5'b0, 2'b01, instr_i[4:2], 2'b01, instr_i[9:7], 3'b000, 2'b01, instr_i[9:7], OPCODE_OP };

380:BS
instr_o = { 7'b0, 2'b01, instr_i[4:2], 2'b01, instr_i[9:7], 3'b110, 2'b01, instr_i[9:7], OPCODE_OP };

366:BS
illegal_instr_o = 1'b1;

364:BS
instr_o = { { 3{ instr_i[12] } }, instr_i[4:3], instr_i[5], instr_i[2], instr_i[6], 4'b0, 5'h02, 3'b000, 5'h02, OPCODE_OP_IMM };

(instr_i[11:7] == 5'h02) ({ instr_i[12], instr_i[6:2] } == 6'b0)

420:CA

instr_i[15:13] instr_i[15:13] instr_i[15:13] instr_i[15:13]

392:CA

instr_i[15:13]

397:CA

instr_i[15:13]

349:IF

(instr_i[12:5] == 8'b0)

409:BS
illegal_instr_o = 1'b1;

(instr_i[11:7] == 5'b0)

392:BL
instr_o = { 7'b0, instr_i[6:2], instr_i[11:7], 3'b001, instr_i[11:7], OPCODE_OP_IMM };

397:BL
instr_o = { 4'b0, instr_i[3:2], instr_i[12], instr_i[6:4], 2'b00, 5'h02, 3'b010, instr_i[11:7], OPCODE_LOAD };

!((instr_i[6:2] != 5'b0)) (instr_i[6:2] != 5'b0)