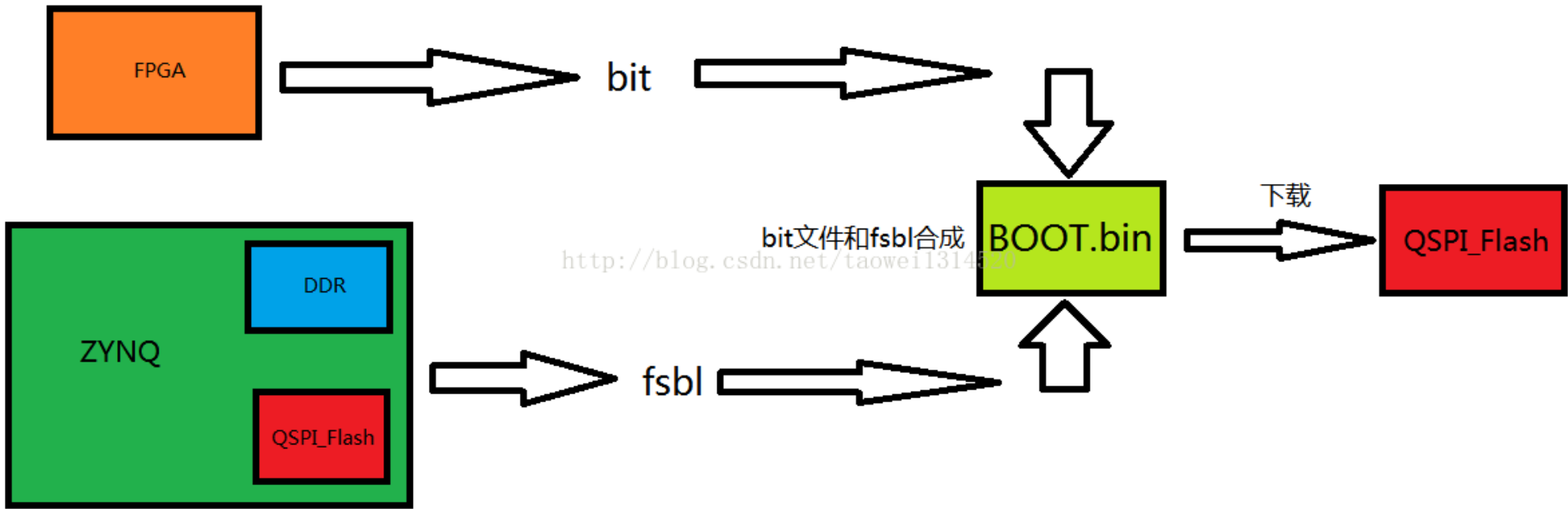


# MYIR-ZYNQ7000系列-zturn教程(9)：将bit文件固化到QSPI\_Flash

原创 虚无缥缈vs威武 最后发布于2017-11-30 13:50:35 阅读数 4059 ☆ 收藏

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1。

我们用FPGA最后生成的是二进制bit文件，bit文件下载到开发板是在内存里运行如果断电程序会丢失。如果我们用FPGA生成的bit文件和zynq核生成BOOT.bin文件，这个zynq核配置了DDR核QSPI\_Flash,然后将这个BOOT.bin文件下载到QSPI\_Flash让开发板运行在QSPI\_Flash启动模式就能让开发板程序也不会丢失。如下图所示



Step1 这是已经新建好的一个vivado 流水灯的fpga工程

👍  
2

🔗

展开

💬  
5

☆ 合成一个程序也不会

📱

<

>

赏

led\_flash - [D:/word\_project/2017-11-9/led\_flash/led\_flash.xpr] - Vivado 2017.1

FileEditFlowToolsWindowLayoutViewHelpQuick Access

Flow Navigator

PROJECT MANAGER

Settings

Add Sources

Language Templates

IP Catalog

IP INTEGRATOR

Create Block Design

Open Block Design

Generate Block Design

SIMULATION

Run Simulation

RTL ANALYSIS

Open Elaborated Design

SYNTHESIS

Run Synthesis

Open Synthesized Design

IMPLEMENTATION

Run Implementation

Open Implemented Design

PROGRAM AND DEBUG

Generate Bitstream

Open Hardware Manager

PROJECT MANAGER - led\_flash

Sources

Design Sources (1)

user\_led (user\_led.v)

Constraints (1)

constrs\_1 (1)

led\_flash.xdc

Simulation Sources (1)

sim\_1 (1)

HierarchyLibrariesCompile Order

Source File Properties

user\_led.v

Enabled

Location: D:/word\_project/2017-11-9/led\_flash/led\_flash.s

Type: Verilog

Library: xil\_defaultlib

Size: 1.5 KB

GeneralProperties

Tcl ConsoleMessagesLogReportsDesign Runs

Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start
synth_1	constrs_1	Not started													
impl_1	constrs_1	Not started													

Project Summary

user\_led.v

led\_flash.xdc

21

22

23 module user\_led

24 (

25 i\_clk,

26 i\_rst\_n,

27 led

28 );

29

30 input i\_clk;

31 input i\_rst\_n;

32 output [2:0] led;

33

34 reg [25:0] cnt;

35 reg [2:0] led\_r;

36 reg [3:0] count;

37

38 always@(posedge i\_clk or negedge i\_rst\_n)//1s的计数led有点慢这里用的是0.5s计数

39 if(i\_rst\_n==1'b0)

40 cnt <= 26'd0;

41 else if(cnt==26'd4999999)

42 cnt <= 26'd0;

43 else

44 cnt <= cnt + 1'b1;

45

46 always@(posedge i\_clk or negedge i\_rst\_n)

47 if(i\_rst\_n==1'b0)

48 count <= 4'd0;

2

5

赏

Step2 点击Flow Navigator 下的Create Block Desion新建一个Block Design

led\_flash - [D:/word\_project/2017-11-9/led\_flash/led\_flash.xpr] - Vivado 2017.1

File Edit Flow Tools Window Layout View Help Quick Access

Flow Navigator

PROJECT MANAGER - led\_flash

Sources

- Design Sources (1)
  - user\_led (user\_led.v)
- Constraints (1)
  - constrs\_1 (1)
    - led\_flash.xdc
- Simulation Sources (1)
  - sim\_1 (1)

Hierarchy Libraries Compile Order

Source File Properties

user\_led.v

Enabled

Location: D:/word\_project/2017-11-9/led\_flash/led\_flash.s

Type: Verilog

Library: xil\_defaultlib

Size: 1.5 KB

General Properties

Project Summary

user\_led.v led\_flash.xdc

```
21
22
23 module user_led
24 (
25     i_clk,
26     i_rst_n,
27     led
28 );
29
30 input i_clk;
31 input i_rst_n;
32 output [2:0] led;
33
34 reg [25:0] cnt;
35 reg [2:0] led_r;
36 reg [3:0] count;
37
38 always@(posedge i_clk or negedge i_rst_n) //1s的计数led有点慢这里用的是0.5s计数
39     if(i_rst_n==1'b0)
40         cnt <= 26'd0;
41     else if(cnt==26'd4999999)
42         cnt <= 26'd0;
43     else
44         cnt <= cnt + 1'b1;
45
46 always@(posedge i_clk or negedge i_rst_n)
47     if(i_rst_n==1'b0)
48         count <= 4'd0;
```

Tcl Console Messages Log Reports Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start	El:
synth_1	constrs_1	Not started														
impl_1	constrs_1	Not started														

点击OK

Create Block Design

Please specify name of block design.

Design name: design\_1

Directory: <Local to Project>

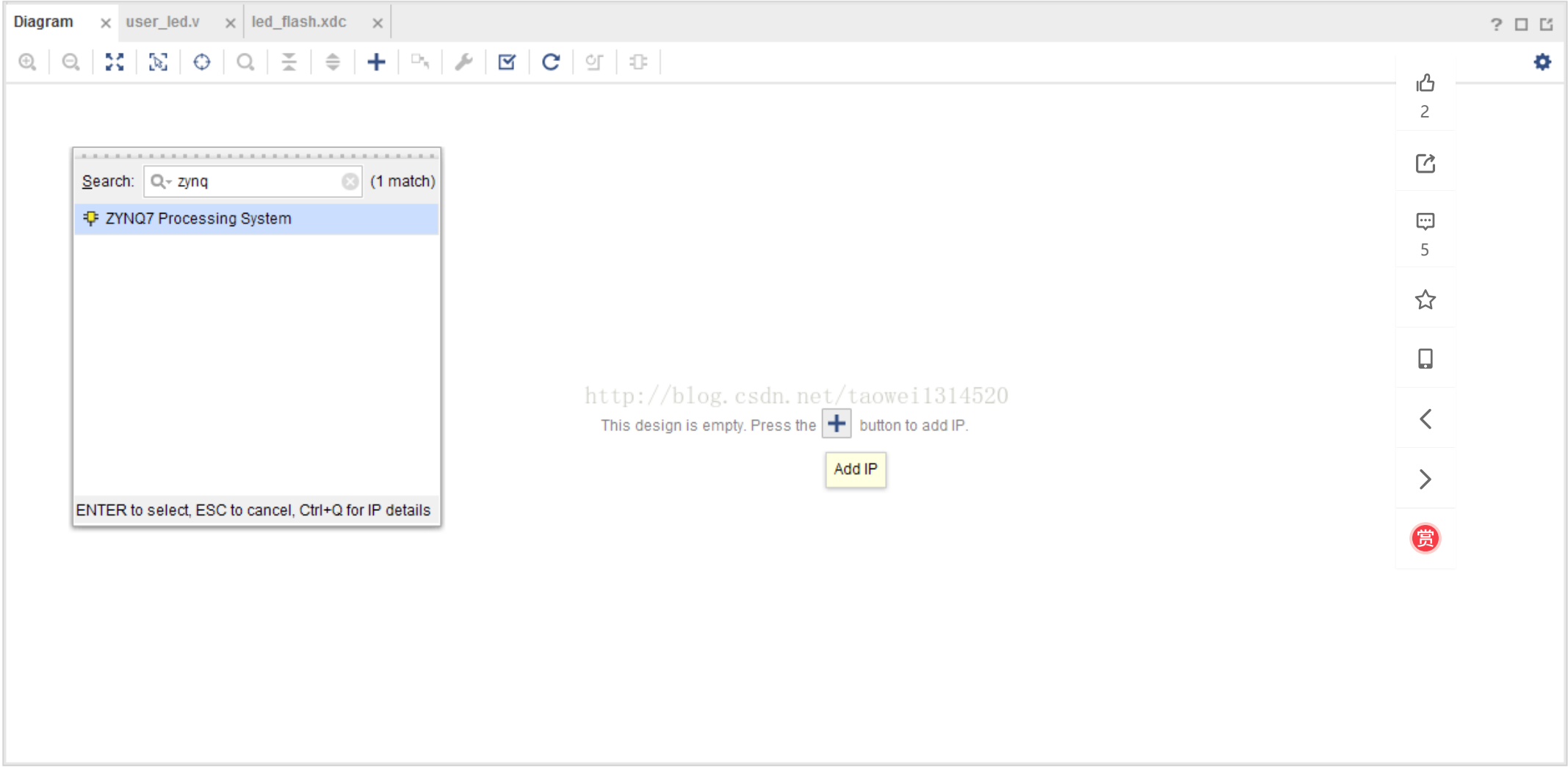
Specify source set: Design Sources

OK Cancel

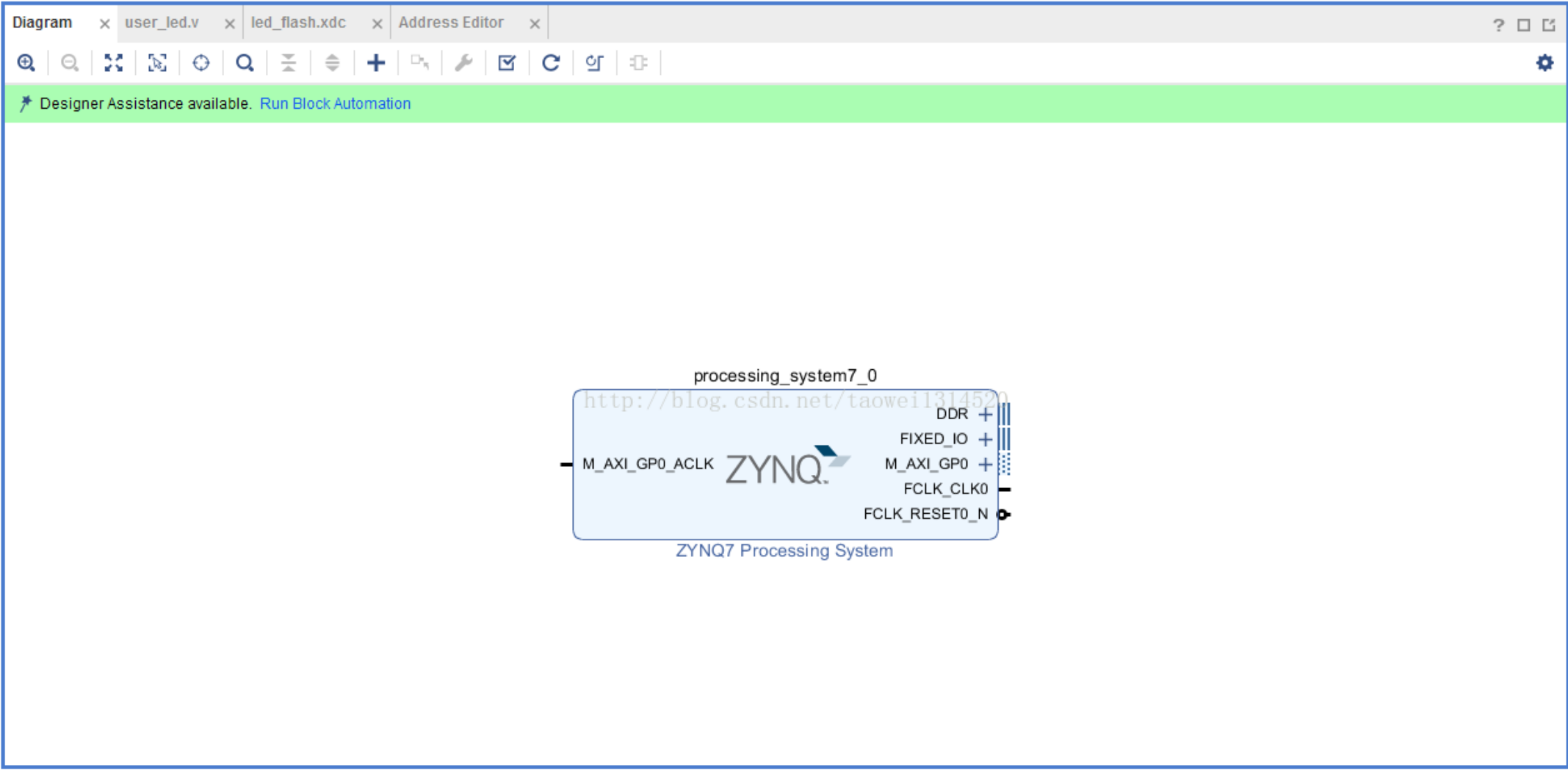
Step3 点击工作区的Add IP添加IP核，在弹出的搜索栏Search中输入zynq，发现可以搜到ZYNQ IP核，双击这个ZYNQ核添加到工作区内



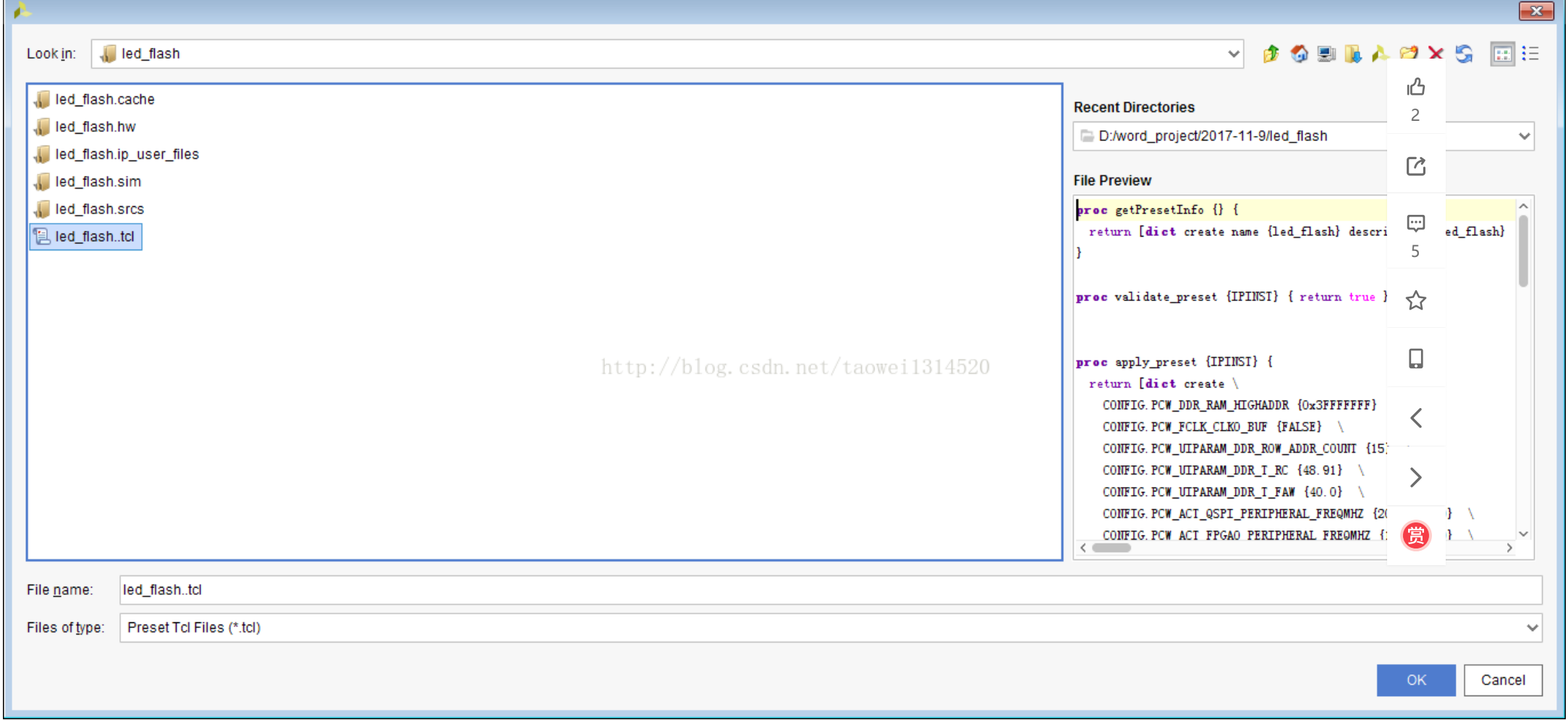
举报



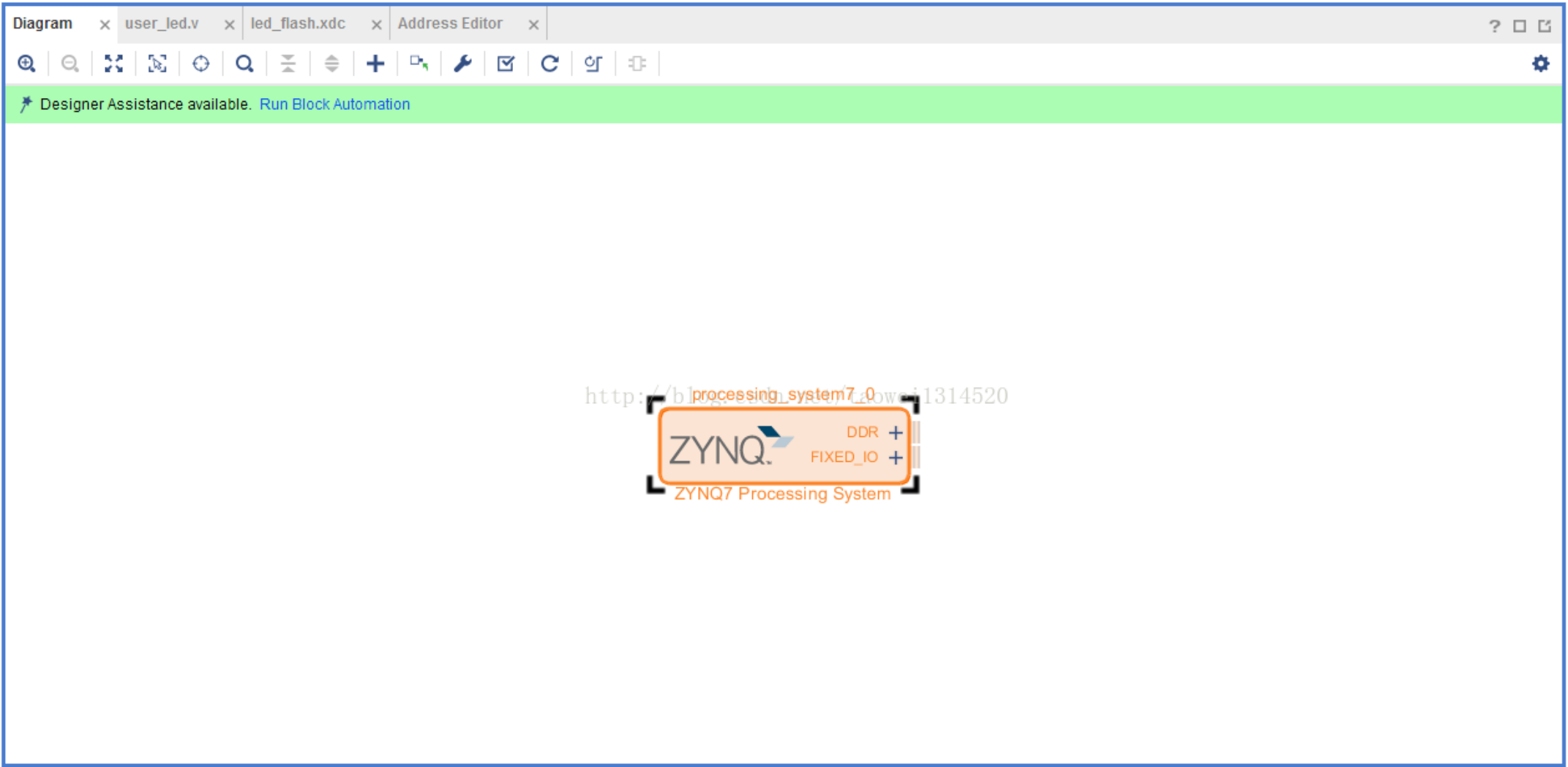
添加好的ZYNQ核如下图所示



Step4 双击ZYNQ IP核添加tcl配置文件,然后点击OK (如果不知道怎样添加请参考hello\_world工程里面有详细步骤)

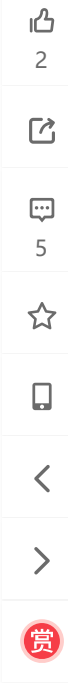
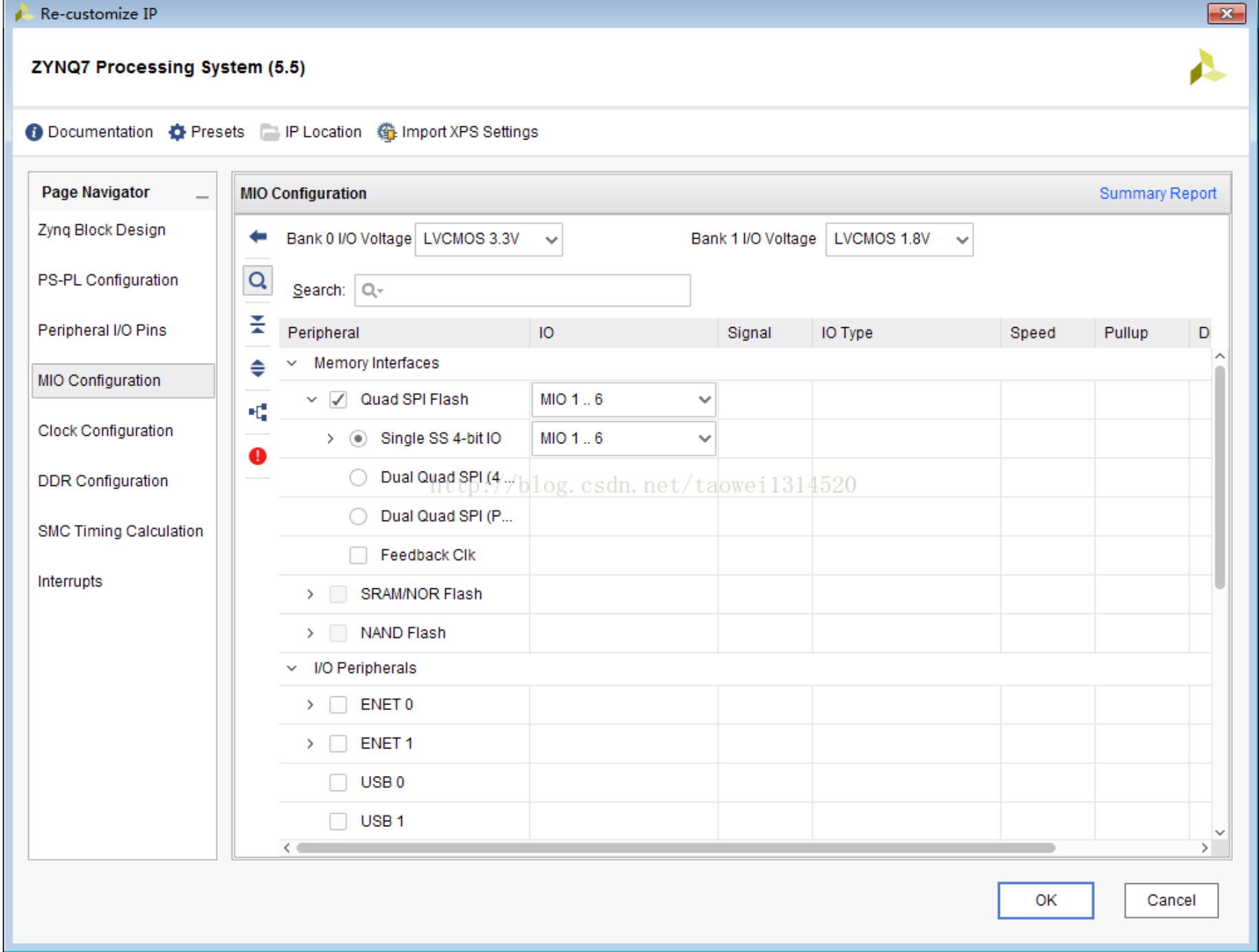


已经配置好的zynq核，如下图所示

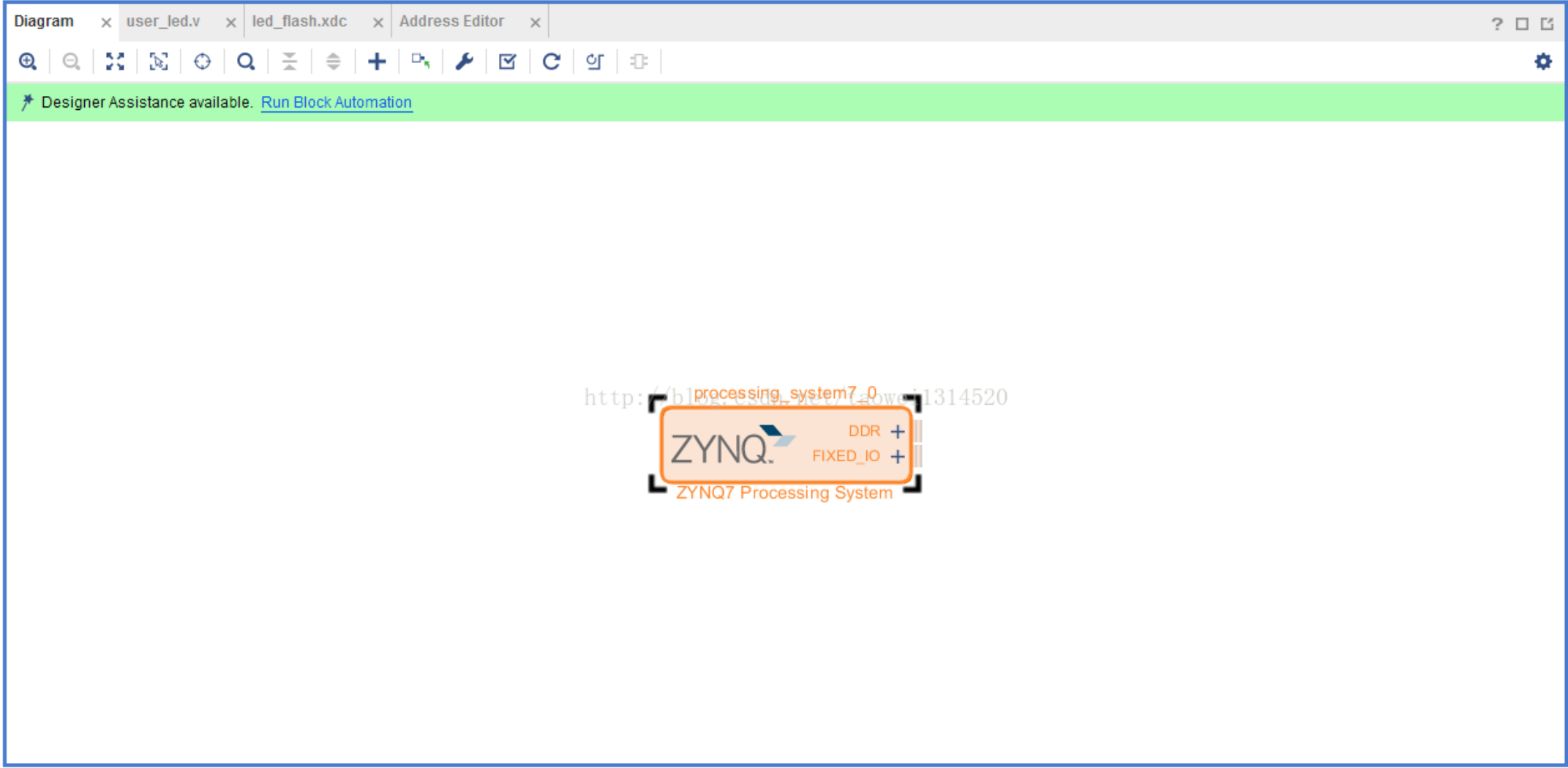


我们配置zynq核时一定要不要忘记配置这个QSPI\_Flash，有很多人没有这里没有配置

QSPI\_Flash导致最后下载到QSPI\_Flash时失败

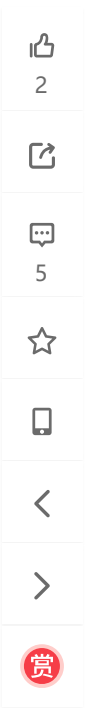
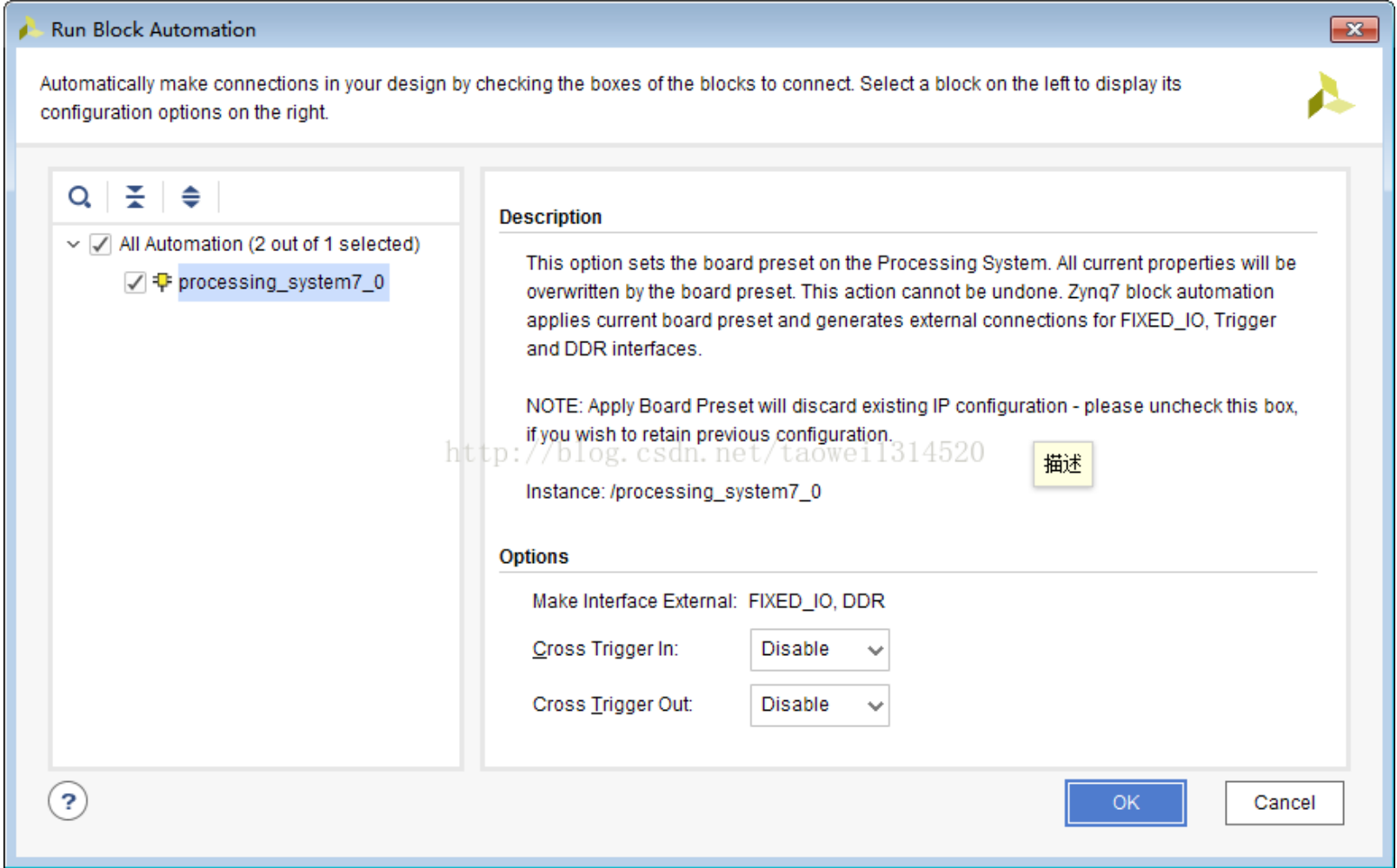


Step 5 点击Run Block Automation进行自动布线

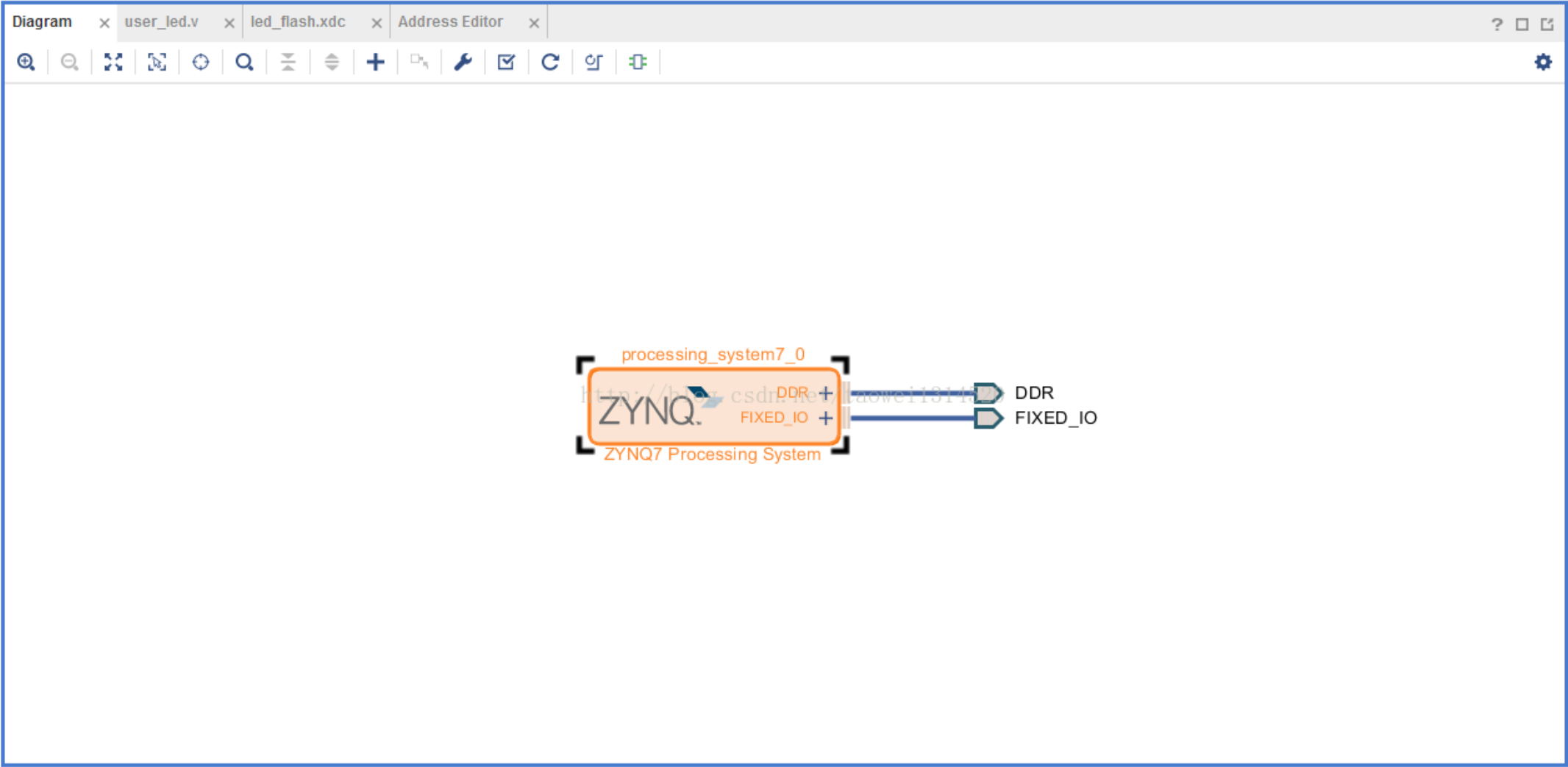


在弹出的对话框中点击OK



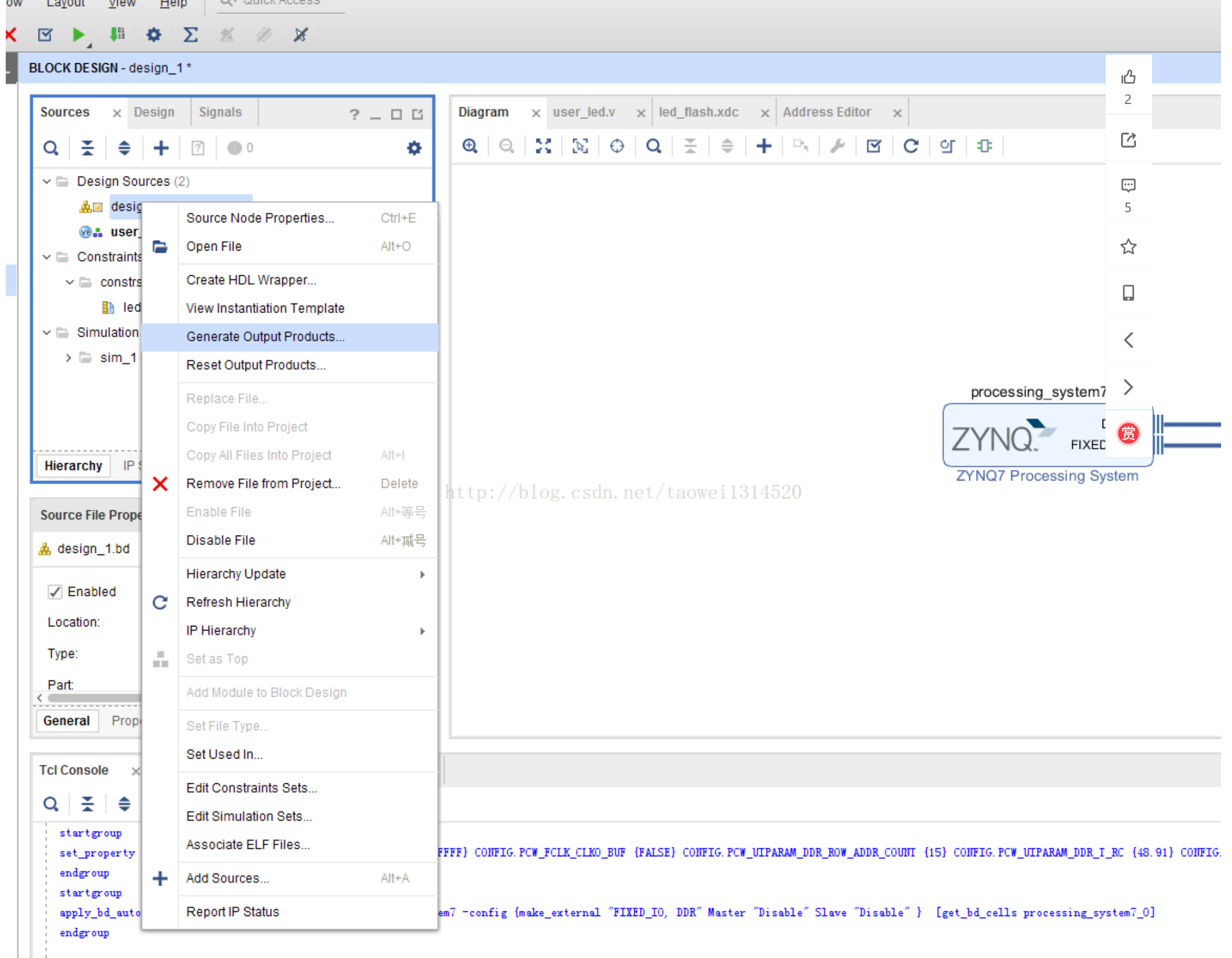


自动布线完成后如下图所示

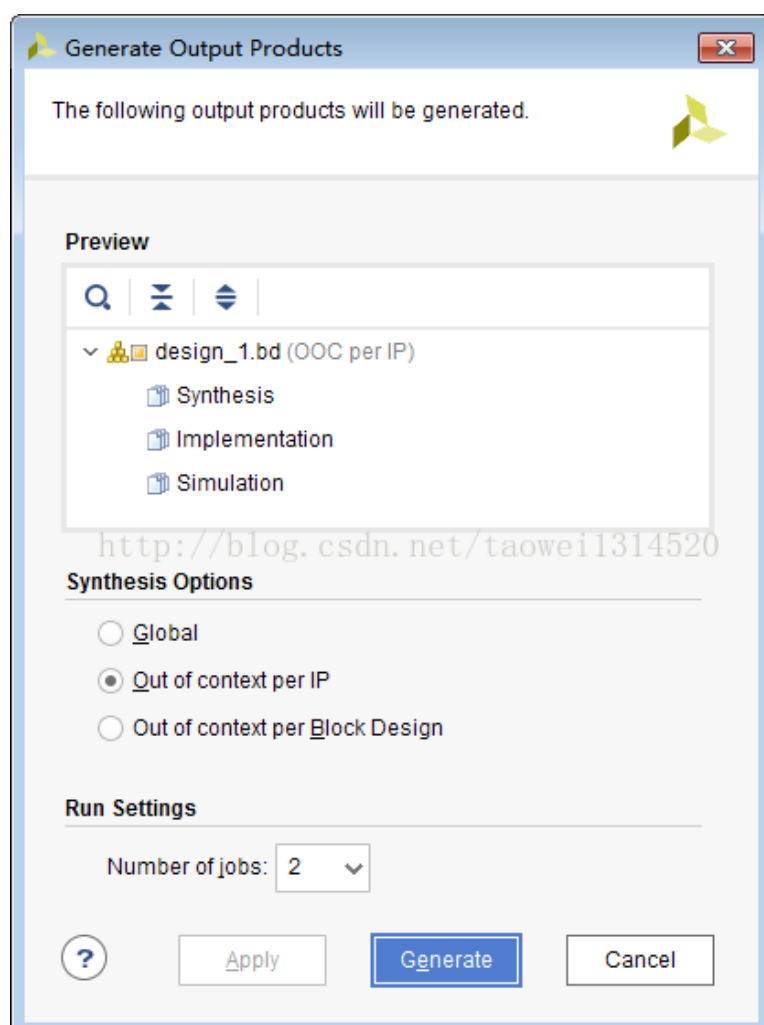


Step6 右击 design\_1->Generate Ouput Products 生成综合文件





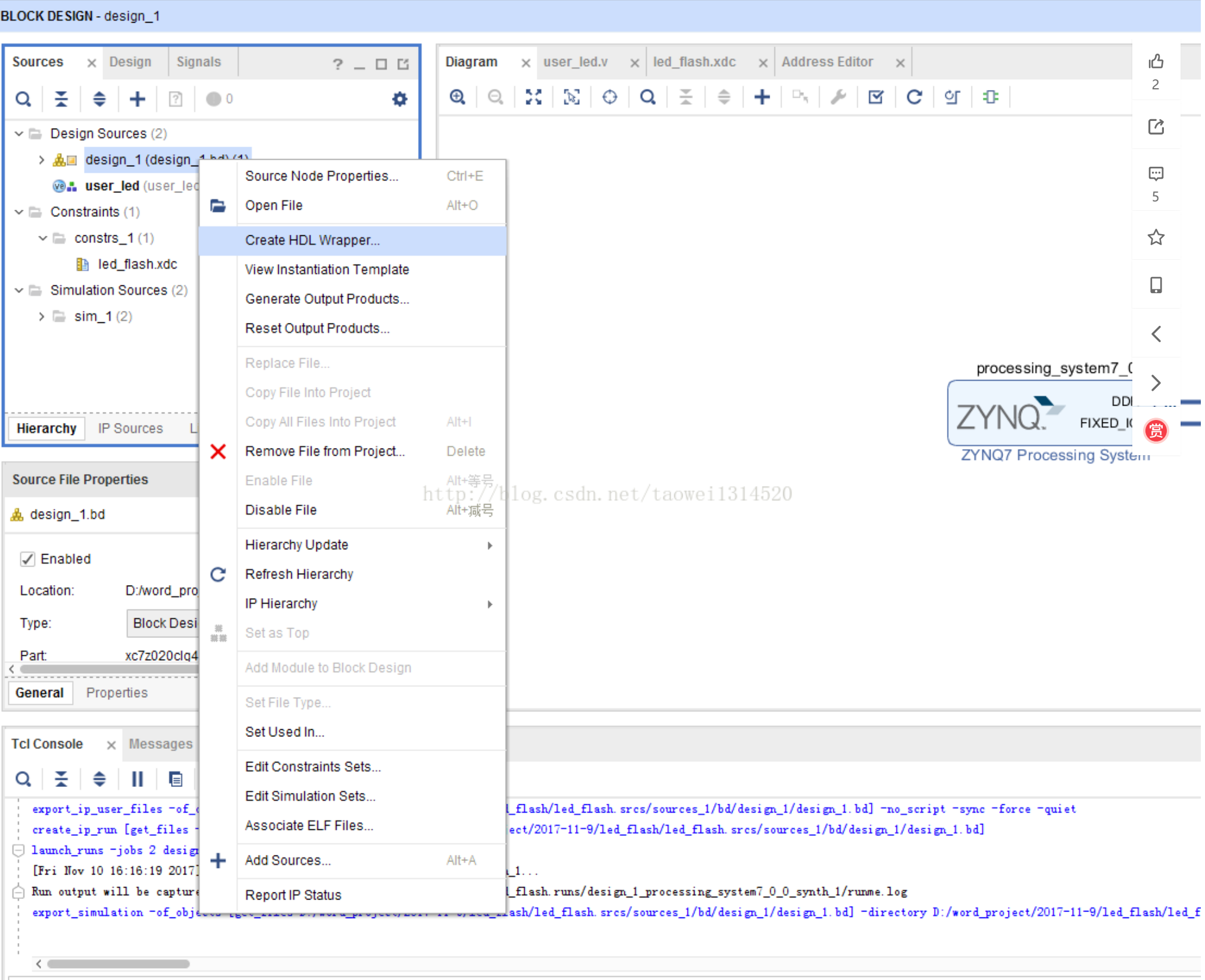
在弹出的对话框中点击Generate



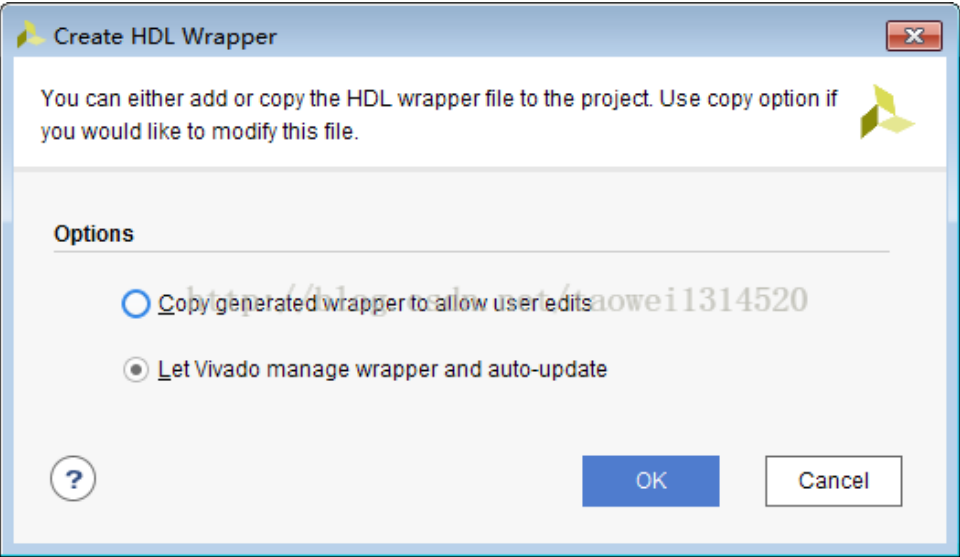
Step7 右击 design\_1->Create HDL Wrapper 生成顶层文件







在弹出的对话框中点击OK



顶层文件生成后如下图所示

BLOCK DESIGN - design\_1

Sources x Design Signals ? \_ □ □

Design Sources (2)

- design\_1\_wrapper (design\_1\_wrapper.v) (1)
  - user\_led (user\_led.v)

Constraints (1)

- constrs\_1 (1)
  - led\_flash.xdc

Simulation Sources (2)

- sim\_1 (2)

Hierarchy IP Sources Libraries Compile Order

Diagram x user\_led.v x led\_flash.xdc x Address Editor x

Source File Properties ? \_ □ □ x

design\_1.bd

Enabled

Location: D:/word\_project/2017-11-9/led\_flash/led\_flash

Type: Block Designs

Part: xc7z020clq400-1

General Properties

Tcl Console x Messages Log Reports Design Runs

Run output will be captured here: D:/word\_project/2017-11-9/led\_flash/led\_flash.runs/design\_1\_processing\_system7\_0\_0\_synth\_1/runme.log

export\_simulation -of\_objects [get\_files D:/word\_project/2017-11-9/led\_flash/led\_flash.srcs/sources\_1/bd/design\_1/design\_1.bd] -directory D:/word\_project/2017-11-9/led\_flash/led\_flash.runs

make\_wrapper -files [get\_files D:/word\_project/2017-11-9/led\_flash/led\_flash.srcs/sources\_1/bd/design\_1/design\_1.bd] -top

add\_files -norecurse D:/word\_project/2017-11-9/led\_flash/led\_flash.srcs/sources\_1/bd/design\_1/hdl/design\_1\_wrapper.v

update\_compile\_order -fileset sources\_1

update\_compile\_order -fileset sources\_1

processing\_system7

ZYNQ FIXED

ZYNQ7 Processing System

<http://blog.csdn.net/taowei1314520>

将这个流水灯user\_led例化到这个zynq核的顶层文件里，如下图所示

Sources

Design

Signals

?

—

□

✎

0

Design Sources (2)

design\_1\_wrapper (design\_1\_wrapper.v) (1)

design\_1\_i : design\_1 (design\_1.bd) (1)

design\_1 (design\_1.v) (1)

processing\_system7\_0 : design\_1\_proc...

user\_led (user\_led.v)

Constraints (1)

constrs\_1 (1)

led\_flash.xdc

Simulation Sources (2)

sim\_1 (2)

Hierarchy

IP Sources

Libraries

Compile Order

Source File Properties

design\_1\_wrapper.v

Enabled

Location: D:/word\_project/2017-11-9/led\_flash/led\_flash

Type: Verilog

Library: vil\_defaultlib

General

Properties

Diagram

user\_led.v

led\_flash.xdc

Address Editor

design\_1\_wrapper.v \*

64

wire [31:0]DDR\_dq;

65

wire [3:0]DDR\_dqs\_n;

66

wire [3:0]DDR\_dqs\_p;

67

wire DDR\_odt;

68

wire DDR\_ras\_n;

69

wire DDR\_reset\_n;

70

wire DDR\_we\_n;

71

wire FIXED\_IO\_dds\_vrn;

72

wire FIXED\_IO\_dds\_vrp;

73

wire [53:0]FIXED\_IO\_mio;

74

wire FIXED\_IO\_ps\_clk;

75

wire FIXED\_IO\_ps\_porb;

76

wire FIXED\_IO\_ps\_srstb;

77

78

user\_led ul

79

{

80

.i\_clk(i\_clk),

81

.i\_rst\_n(i\_rst\_n),

82

.led(led)

83

);

84

85

86

design\_1 design\_1\_i

87

(.DDR\_addr(DDR\_addr),

88

.DDR\_ba(DDR\_ba),

89

.DDR\_cas\_n(DDR\_cas\_n),

90

.DDR\_ck\_n(DDR\_ck\_n),

91

.DDR\_ck\_p(DDR\_ck\_p),

Tcl Console

Messages

Log

Reports

Design Runs

Run output will be captured here: D:/word\_project/2017-11-9/led\_flash/led\_flash.runs/design\_1\_processing\_system7\_0\_0\_synth\_1/runme.log

export\_simulation -of\_objects [get\_files D:/word\_project/2017-11-9/led\_flash/led\_flash.srcs/sources\_1/bd/design\_1/design\_1.bd] -directory D:/word\_project/2017-11-9/le

make\_wrapper -files [get\_files D:/word\_project/2017-11-9/led\_flash/led\_flash.srcs/sources\_1/bd/design\_1/design\_1.bd] -top

add\_files -norecurse D:/word\_project/2017-11-9/led\_flash/led\_flash.srcs/sources\_1/bd/design\_1/hdl/design\_1\_wrapper.v

update\_compile\_order -fileset sources\_1

update\_compile\_order -fileset sources\_1

Type a Tcl command here

2

5

赏

流水灯user\_led例化到zynq核完成后，如下图所示



BLOCK DESIGN - design\_1

Sources Design Signals ? \_ □ □

Design Sources (1)

- design\_1\_wrapper (design\_1\_wrapper.v) (2)
  - u1 : user\_led (user\_led.v)
    - design\_1\_i : design\_1 (design\_1.bd) (1)

Constraints (1)

- constrs\_1 (1)
  - led\_flash.xdc

Simulation Sources (1)

- sim\_1 (1)

Hierarchy IP Sources Libraries Compile Order

Source File Properties ? \_ □ □ ×

design\_1\_wrapper.v

Enabled

Location: D:\word\_project\2017-11-9\led\_flash\led\_flash

Type: Verilog

Library: vil\_defaultlib

General Properties

Diagram user\_led.v led\_flash.xdc Address Editor design\_1\_wrapper

```

74 wire [3:0]DDR_dqs_p;
75 wire DDR_odt;
76 wire DDR_ras_n;
77 wire DDR_reset_n;
78 wire DDR_we_n;
79 wire FIXED_IO_ddr_vrn;
80 wire FIXED_IO_ddr_vrp;
81 wire [53:0]FIXED_IO_mio;
82 wire FIXED_IO_ps_clk;
83 wire FIXED_IO_ps_por;
84 wire FIXED_IO_ps_srstb;
85 wire i_clk;
86 wire i_rst_n;
87 wire [2:0] led;
88
89
90 user_led u1
91 (
92   .i_clk(i_clk),
93   .i_rst_n(i_rst_n),
94   .led(led)
95 );
96
97
98 design_1 design_1_i
99 (
100   .DDR_addr(DDR_addr),
101   .DDR_ba(DDR_ba),
102   .DDR_cas_n(DDR_cas_n),

```

Tcl Console Messages Log Reports Design Runs ×



Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF
✓ synth_1 (active)	constrs_1	synth_design Complete!								0	
✓ impl_1	constrs_1	write_bitstream Complete!	NA	NA	NA	NA	NA	1.691	0	0	
Out-of-Context Module Runs											
> ✓ design_1		Submodule Runs Complete									

Step8 点击Generate Bitstream产生bit文件



2



5



举报

9/led\_flash/led\_flash.xpr] - Vivado 2017.1

Layout View Help Quick Access

BLOCK DESIGN Generate Bitstream

Sources Design Signals ? \_ □ □

Design Sources (1)

- design\_1\_wrapper (design\_1\_wrapper.v) (2)
  - u1 : user\_led (user\_led.v)
    - design\_1\_i : design\_1 (design\_1.bd) (1)
- Constraints (1)
  - constrs\_1 (1)
    - led\_flash.xdc
- Simulation Sources (1)
  - sim\_1 (1)

Hierarchy IP Sources Libraries Compile Order

Source File Properties ? \_ □ □ ×

design\_1\_wrapper.v

Enabled

Location: D:/word\_project/2017-11-9/led\_flash/led\_flash

Type: Verilog ...

Library: vil\_defaultlib ...

General Properties

Tcl Console Messages Log Reports Design Runs ×

74 wire [3:0]DDR\_dqs\_p;

75 wire DDR\_odt;

76 wire DDR\_ras\_n;

77 wire DDR\_reset\_n;

78 wire DDR\_we\_n;

79 wire FIXED\_IO\_ddr\_vrn;

80 wire FIXED\_IO\_ddr\_vrp;

81 wire [53:0]FIXED\_IO\_mio;

82 wire FIXED\_IO\_ps\_clk;

83 wire FIXED\_IO\_ps\_porlb;

84 wire FIXED\_IO\_ps\_srstb;

85 wire i\_clk;

86 wire i\_rst\_n;

87 wire [2:0] led;

88

89

90 user\_led u1

91 (

92 .i\_clk(i\_clk),

93 .i\_rst\_n(i\_rst\_n),

94 .led(led)

95 );

96

97

98 design\_1 design\_1\_i

99 (.DDR\_addr(DDR\_addr),

100 .DDR\_ba(DDR\_ba),

101 .DDR\_cas\_n(DDR\_cas\_n),

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1461

1462

1463

1464

1465

1466

1467

1468

1469

1470

1471

1472

1473

1474

1475

1476

1477

1478

1479

1480

1481

1482

1483

1484

1485

1486

1487

1488

1489

1490

1491

1492

1493

1494

1495

1496

1497

1498

1499

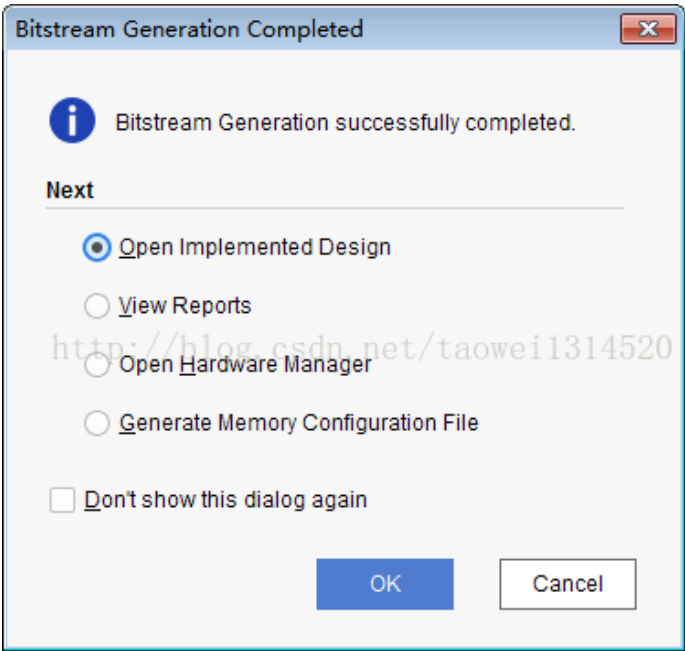
1500


1501


1502


1503


在弹出的对话框中可以看出bit文件生成成功， 点击Cancel关闭对话框





2





5



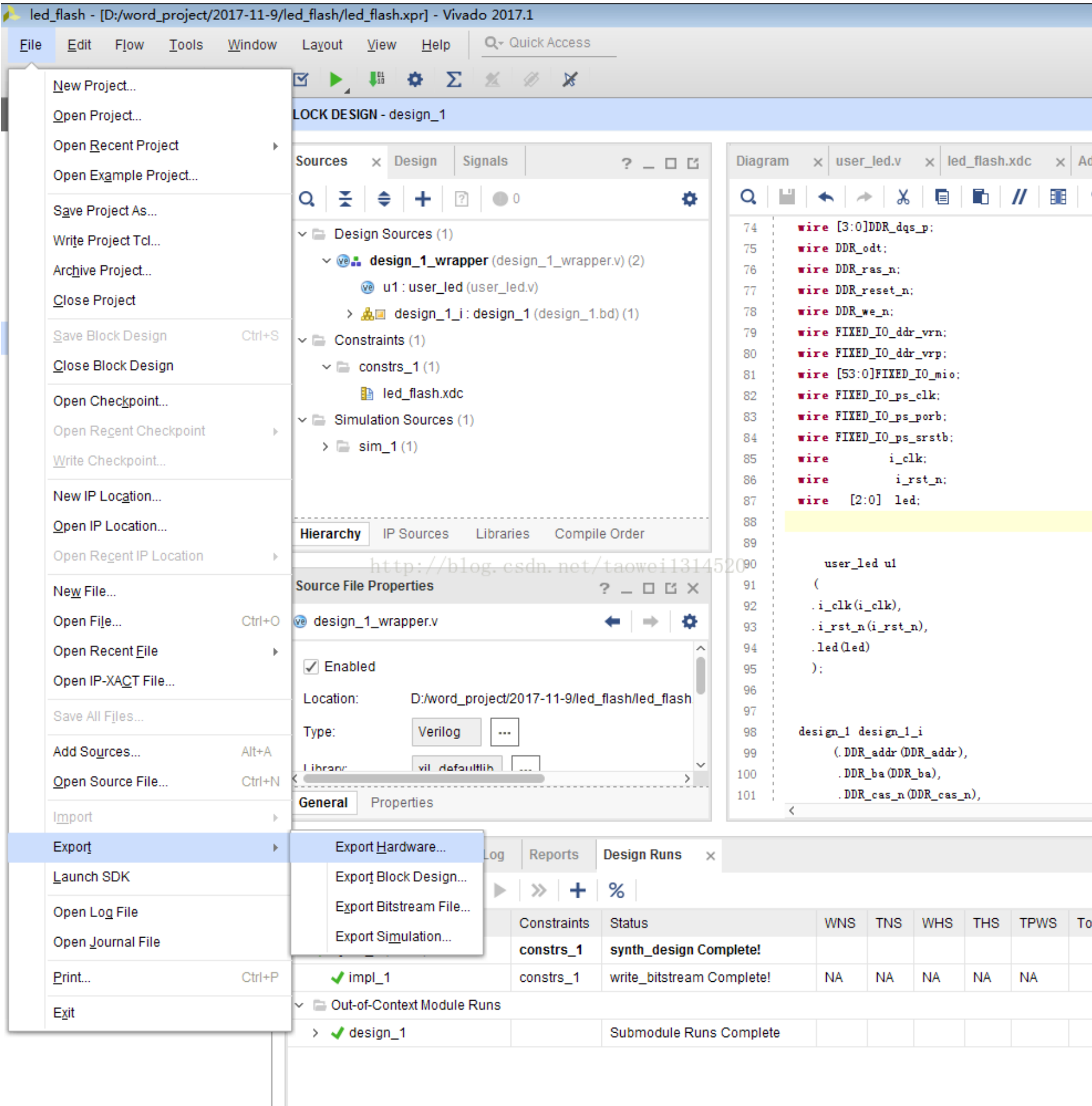








Step9 点击菜单栏上的 File->Export->Export Hardware 导出硬件配置文件

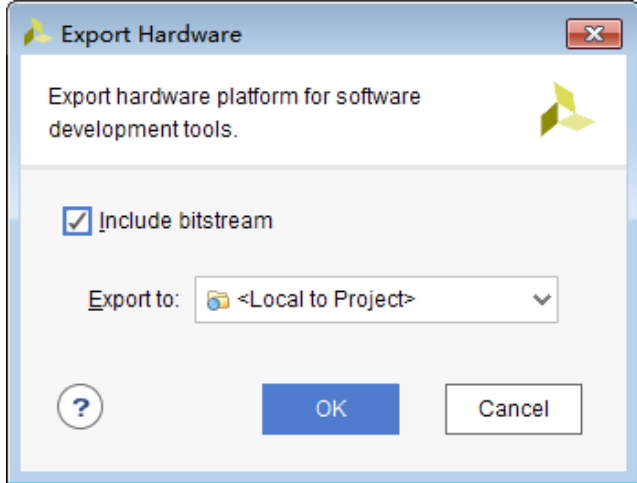


在弹出的对话框中勾选Include bitstream， 然后点击OK

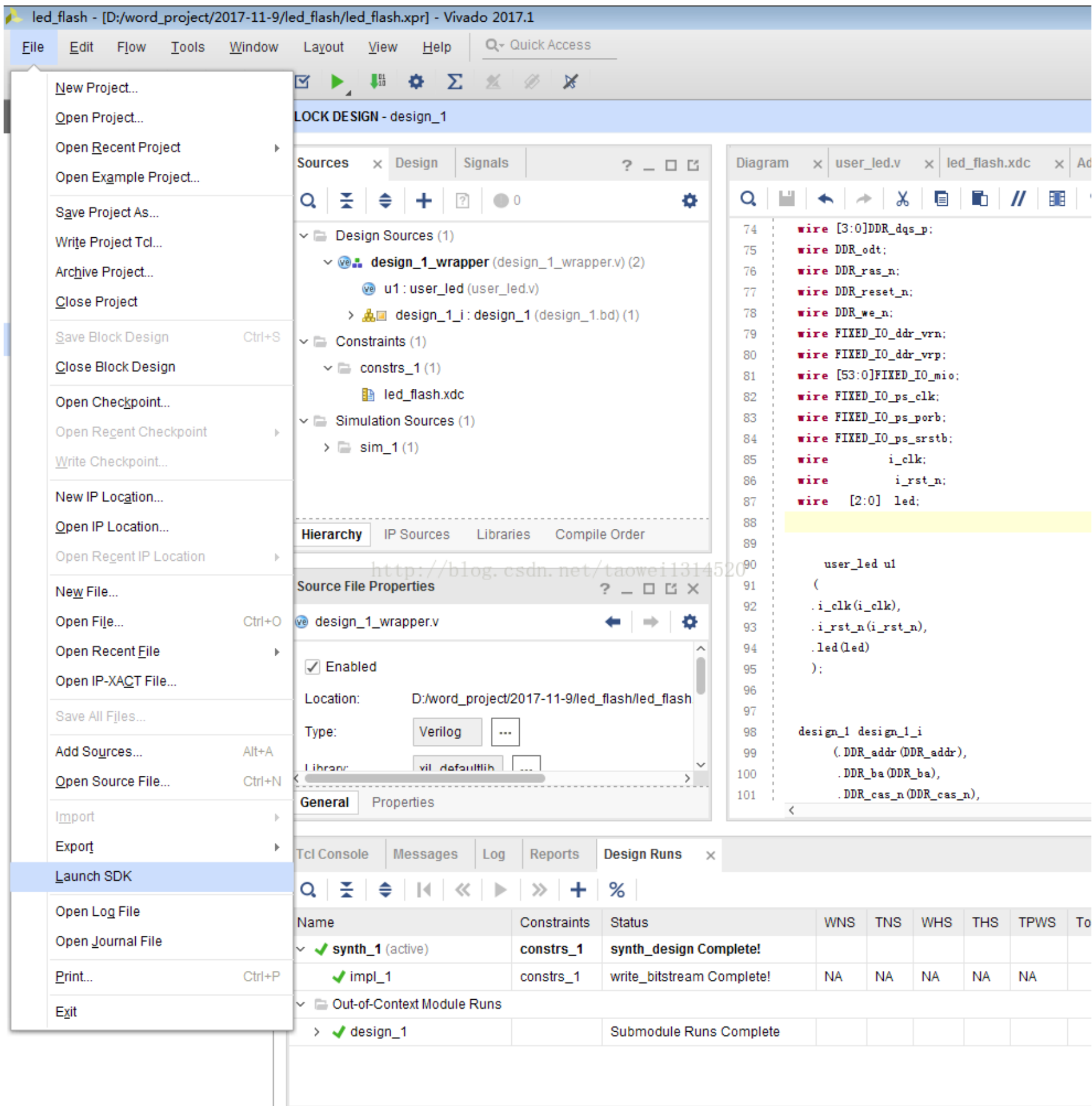


举报

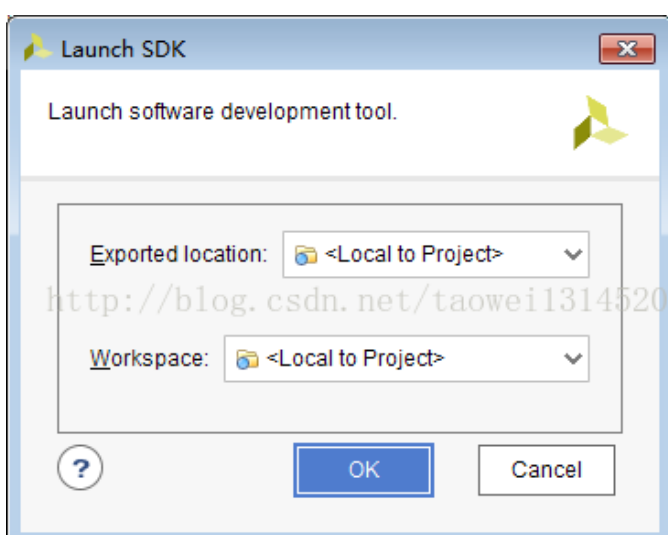




Step10 点击菜单栏上的 File->Launch SDK->OK启动SDK

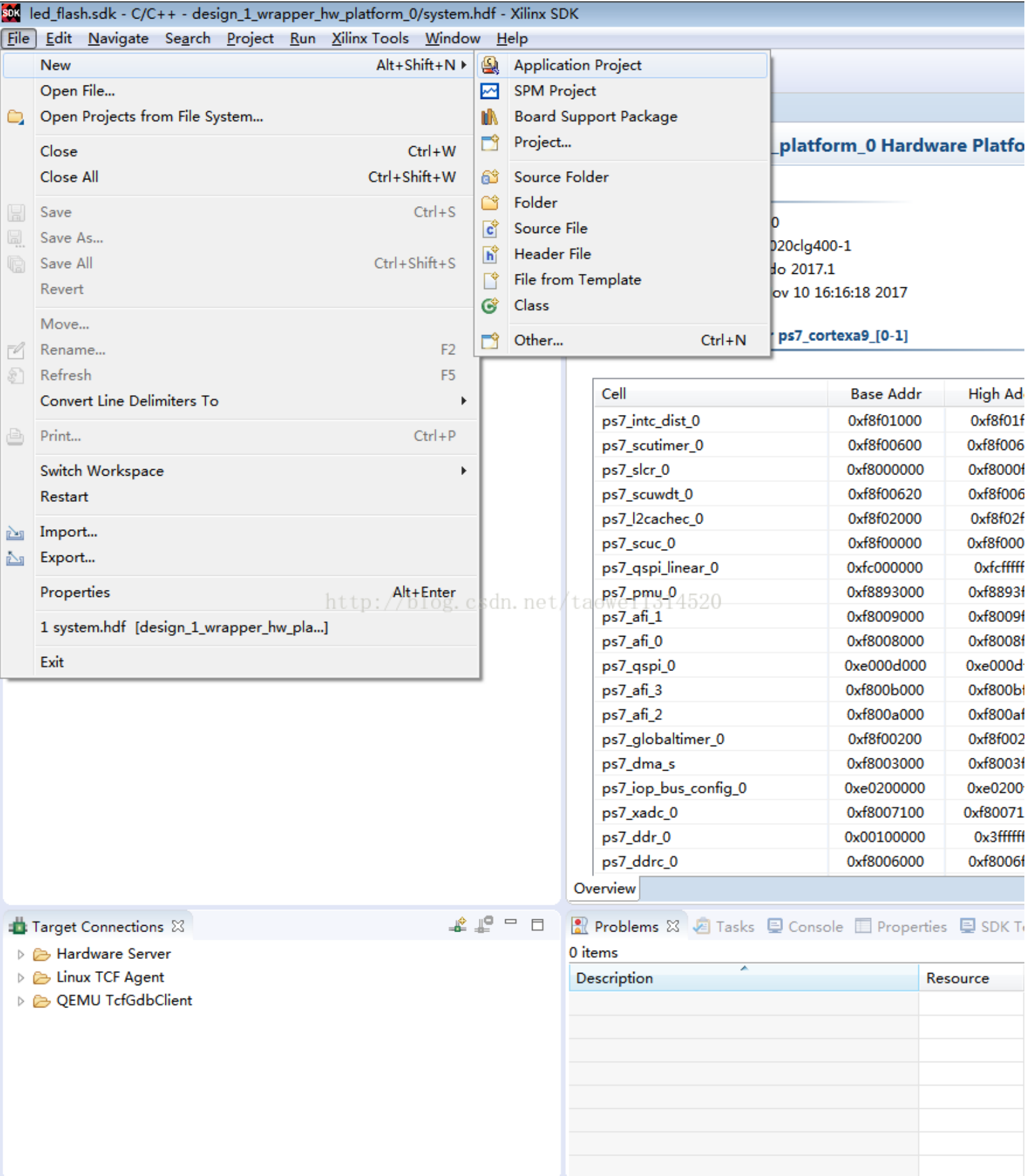


点击OK



举报

Step11 点击菜单栏上的 File->New->Application Project新建工程新建一个fsbl



在弹出的对话框中填写fsbl

👍

2

🔗

💬

5

☆

📱

<

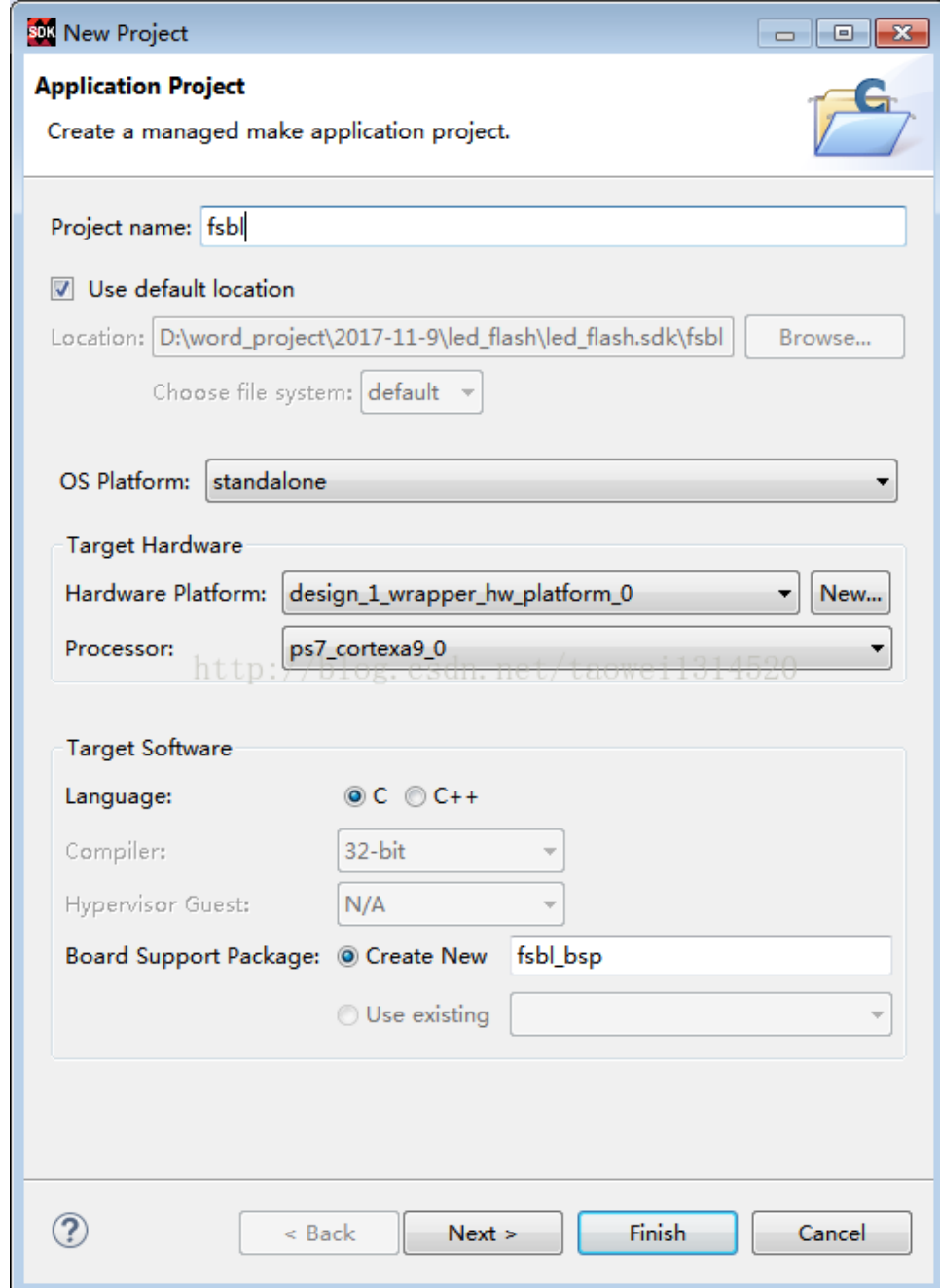
>

赏

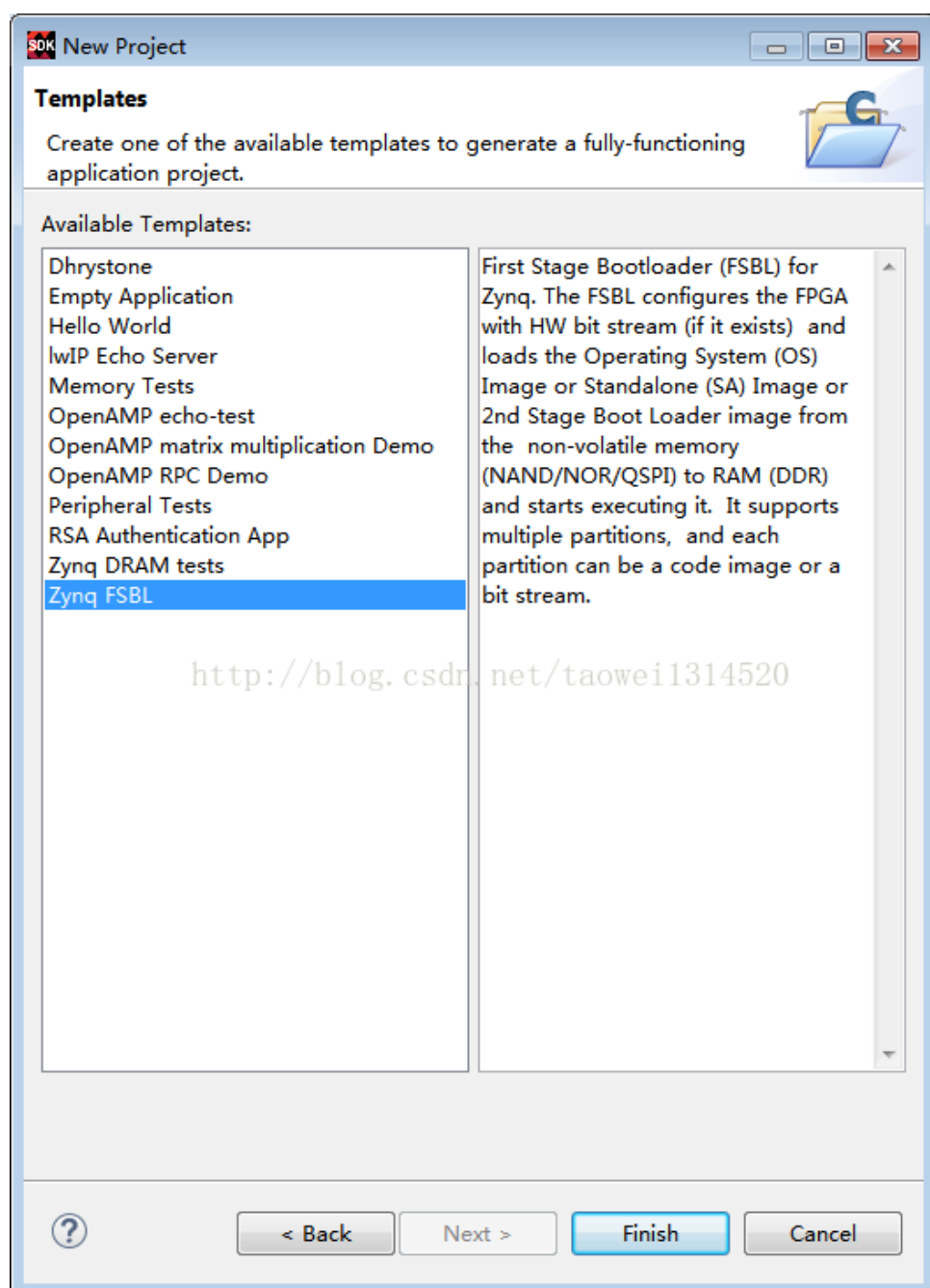
🔊

举报



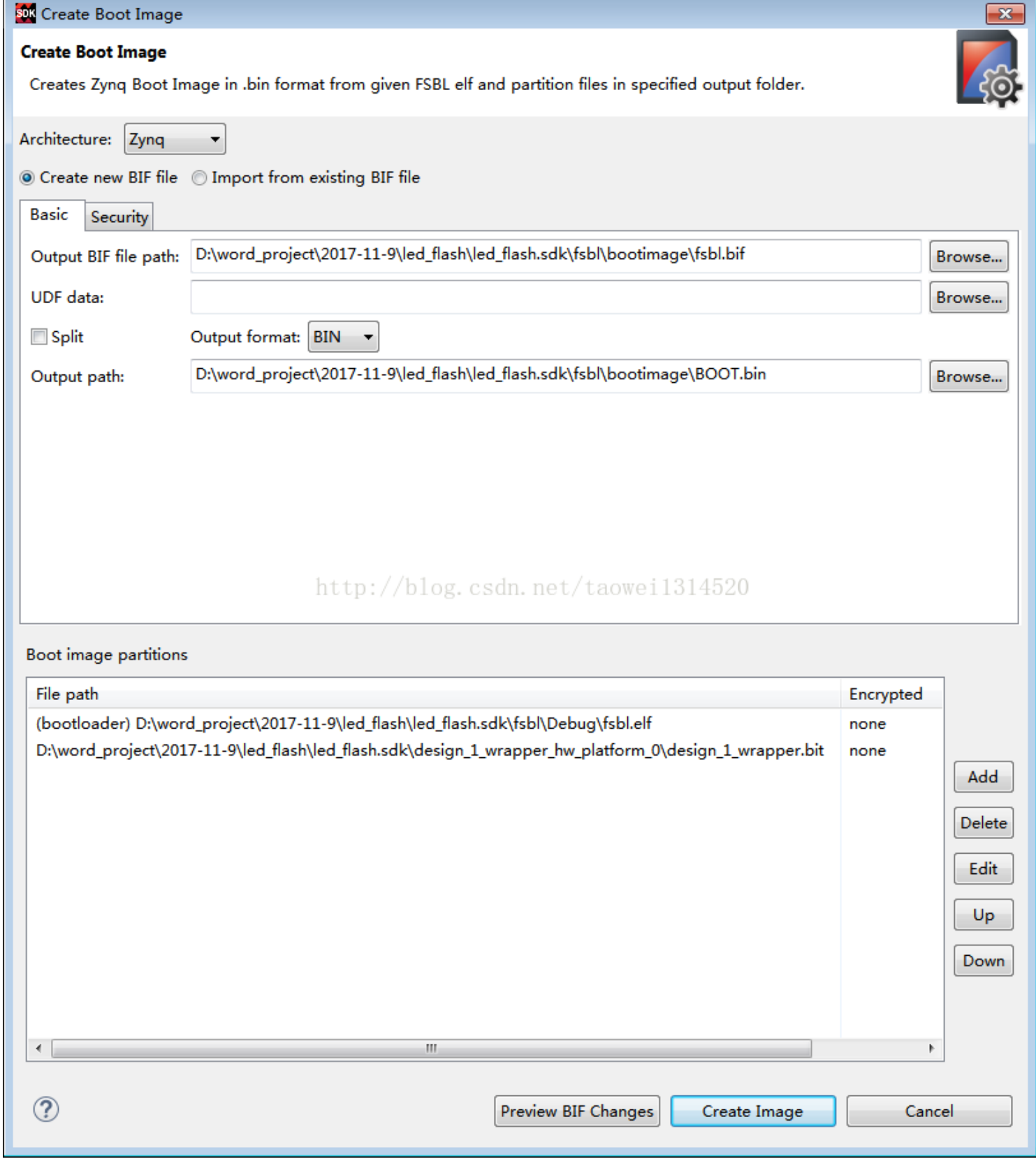


在工程模板中选择Zynq FSBL



Step12 右击 fsbl->Create boot Image, 生成BOOT.bin启动文件

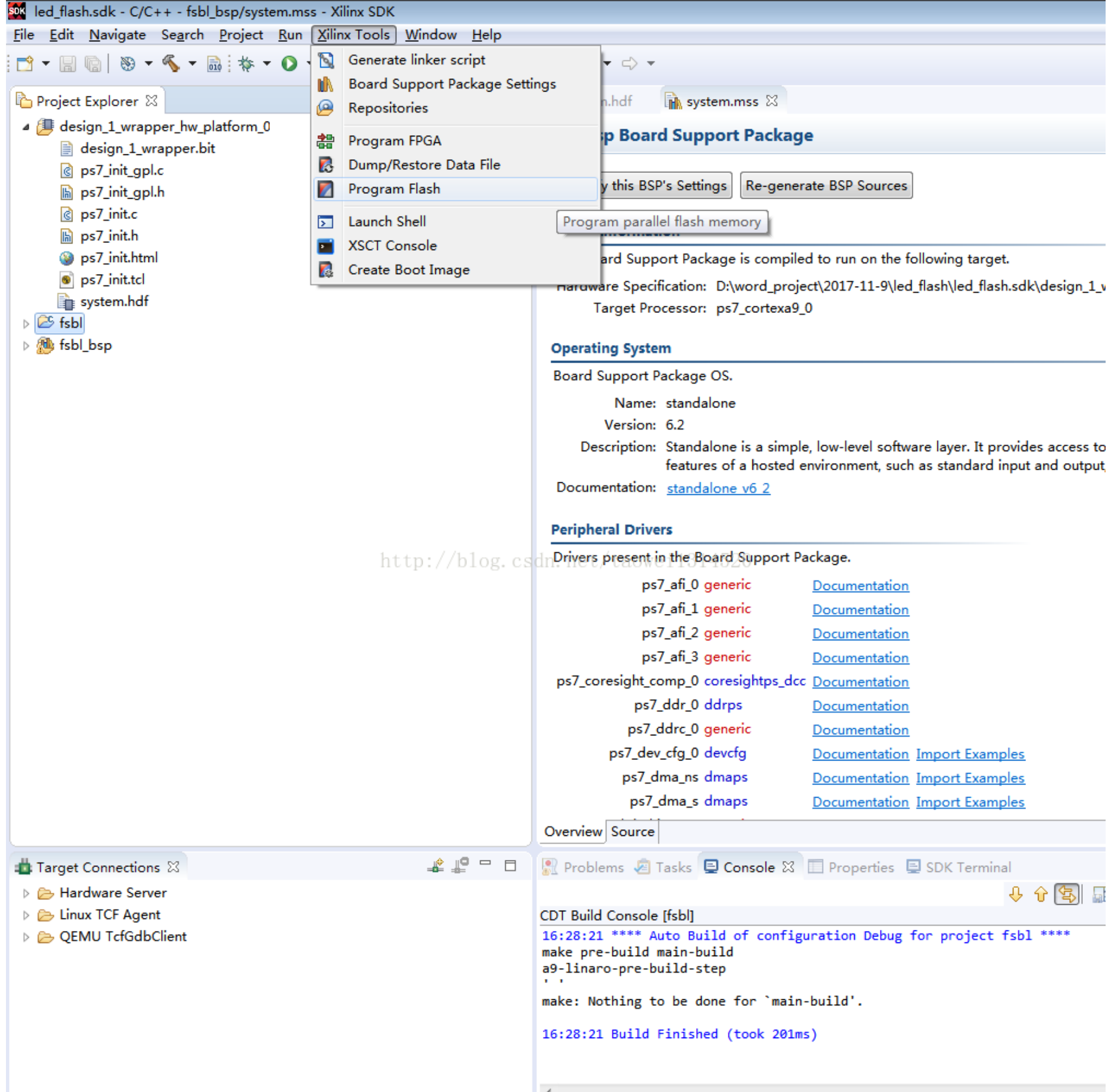




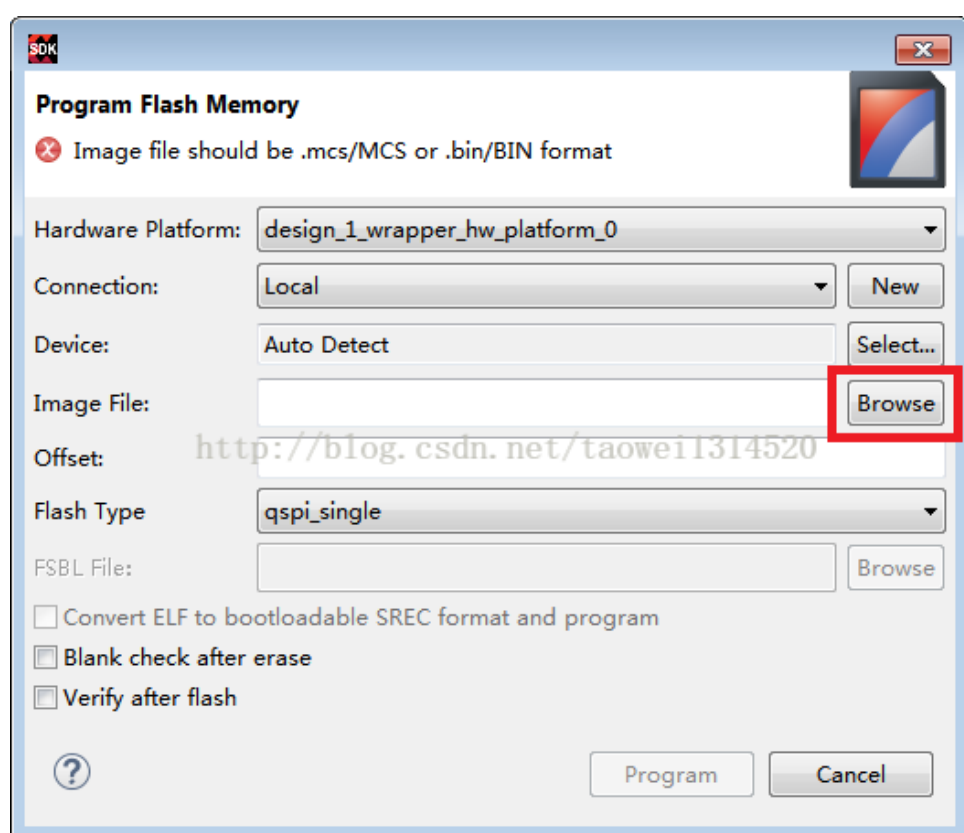
连接开发板电源和jtag下载器，并且将开发板打到SD卡启动模式JP1断开、JP2闭合，如下图所示



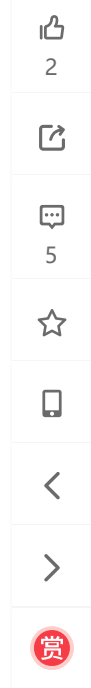
Step13 点击菜单栏上的xilinx Tools-->Program Flash



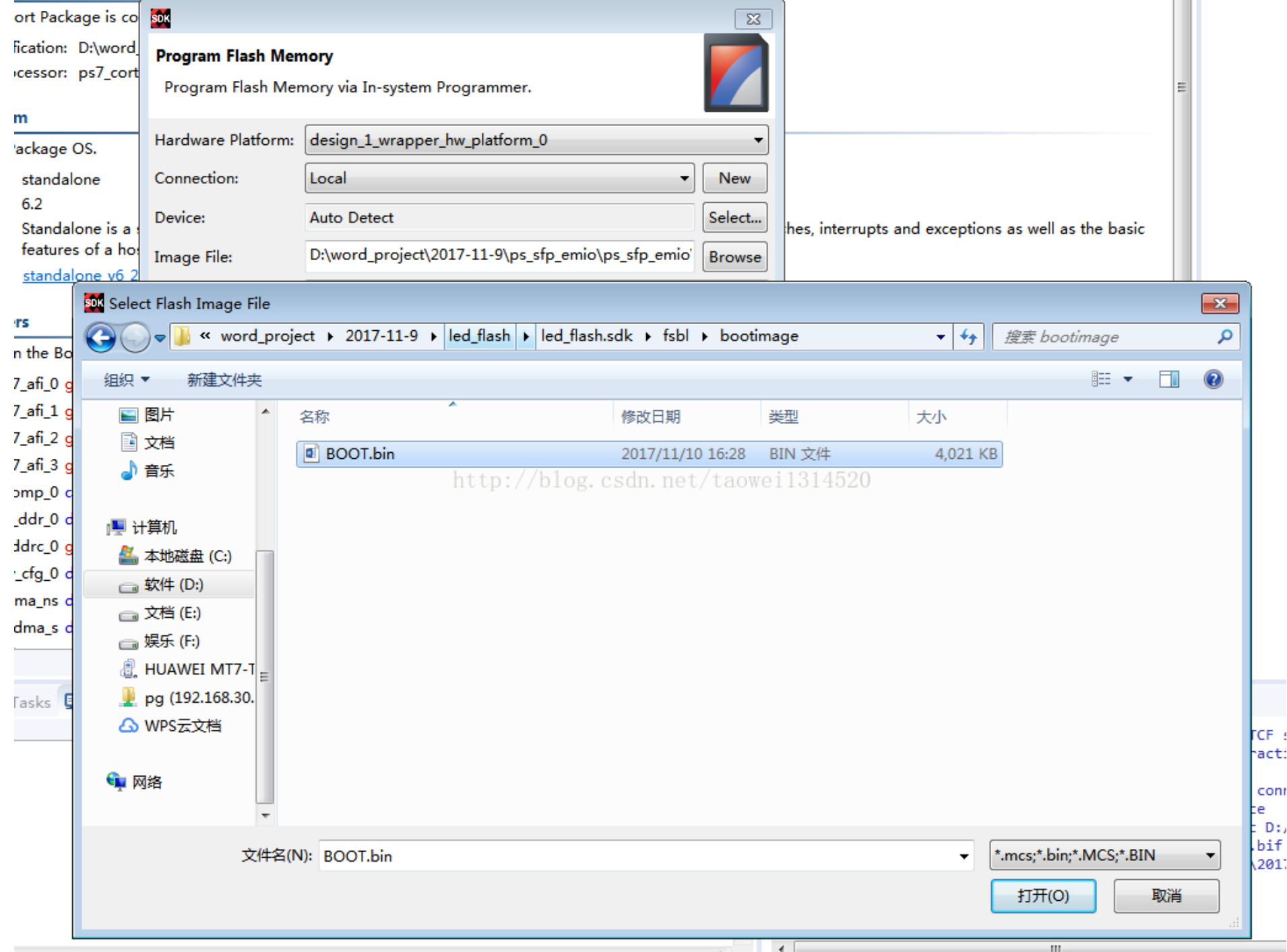
在弹出的对话框中点击Image File 的Browse将我们生成的BOOT.bin文件添加进来



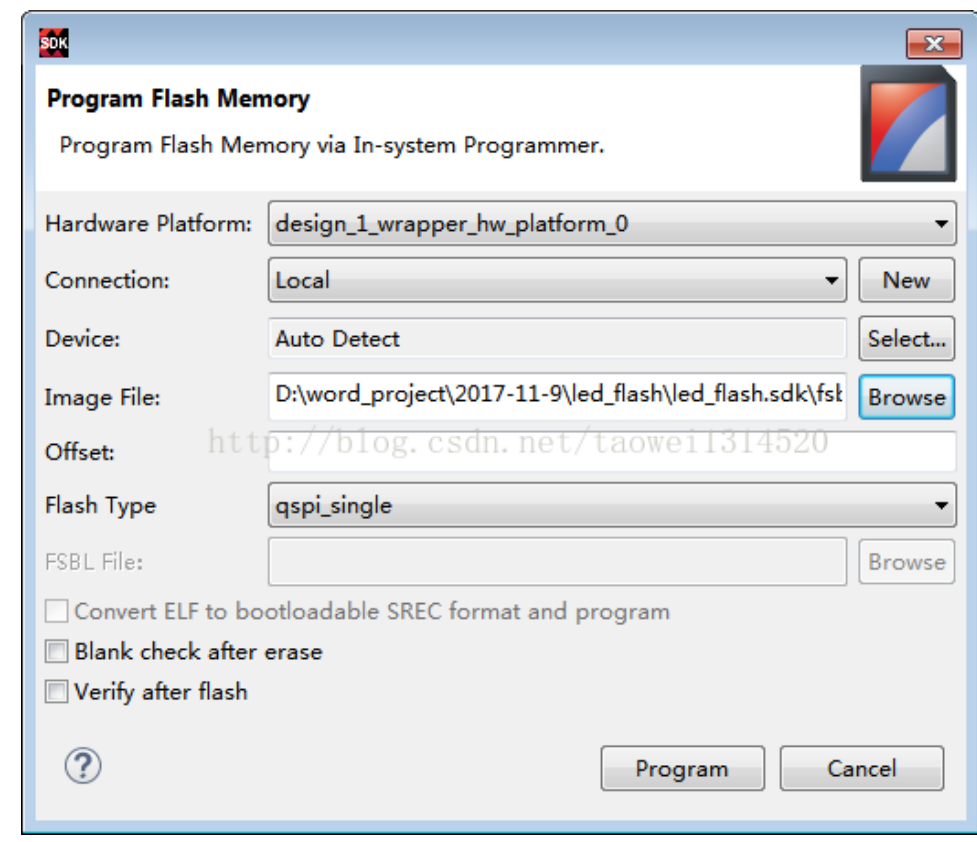
找到我们生成的BOOT.bin文件，然后点击打开将BOOT.bin文件添加进来，如下图所示



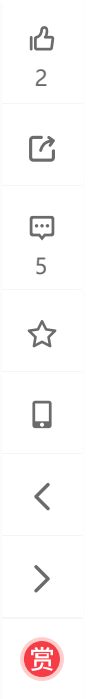




点击Program 将BOOT.bin文件下载到QSPI\_Flash



BOOT.bin文件正在往QSPI\_Flash中下载，如下图所示



system.hdf

system.mss

fsbl\_bsp Board Support Package

Modify this BSP's Settings

Re-generate BSP Sources

Target Information

This Board Support Package is compiled to run on the following target.  
Hardware Specification: D:\word\_project\2017-11-9\led\_flash\led\_flash.sdk\design\_1\_wrapper\_hw\_platform\_0\system.hdf  
Target Processor: ps7\_cortexa9\_0

Operating System

Board Support Package OS.  
Name: standalone  
Version: 6.2  
Description: Standalone is a  
features of a ho  
Documentation: [standalone v6.2](#)

Peripheral Drivers

Drivers present in the Board Support Package:  
ps7\_afi\_0 generic  
ps7\_afi\_1 generic  
ps7\_afi\_2 generic  
ps7\_afi\_3 generic  
ps7\_coresight\_comp\_0 coresightps\_dcc  
ps7\_ddr\_0 ddrps  
ps7\_ddrc\_0 generic  
ps7\_dev\_cfg\_0 devcfg  
ps7\_dma\_ns dmaps  
ps7\_dma\_s dmaps

Overview

Source

SDK Progress Information

Performing programming flash...

Preparing for programming flash

Cancel

Details >>

Problems

Tasks

Console

Properties

SDK Terminal

Program Flash

\*\*\*\*\* Xilinx Program Flash  
\*\*\*\*\* Program Flash v2017.1 (64-bit)  
\*\*\*\*\* SW Build 1846317 on Fri Apr 14 18:55:03 MDT 2017  
\*\*\*\*\* Copyright 1986-2017 Xilinx, Inc. All Rights Reserved.  
  
Connecting to hw\_server @ TCP:127.0.0.1:3121  
  
WARNING: Failed to connect to hw\_server at TCP:127.0.0.1:3121  
Attempting to launch hw\_server at TCP:127.0.0.1:3121

SDK Log

16:25:37 INFO : Registering c  
16:25:38 INFO : Launching XS  
16:25:43 INFO : XSCT server f  
16:25:43 INFO : Successfully  
16:25:46 INFO : Successfully  
16:25:46 INFO : Processing co  
16:28:20 INFO : Invoking Boot  
16:28:20 INFO : Creating new  
16:28:21 INFO : Bootgen comm

2

5

赏

BOOT.bin文件往QSPI\_Flash里下载成功，如下图所示

举报

led\_flash.sdk - C/C++ - fsbl\_bsp/system.mss - Xilinx SDK

File Edit Navigate Search Project Run Xilinx Tools Window Help

Project Explorer

design\_1\_wrapper\_hw\_platform\_0

cache

design\_1\_wrapper.bitps7\_init\_gpl.cps7\_init\_gpl.hps7\_init.cps7\_init.hps7\_init.htmlps7\_init.tclsystem.hdf

fsbl

fsbl\_bsp

system.hdfsystem.mss

fsbl\_bsp Board Support Package

Modify this BSP's SettingsRe-generate BSP Sources

Target Information

This Board Support Package is compiled to run on the following target.

Hardware Specification: D:\word\_project\2017-11-9\led\_flash\led\_flash.sdk\design\_1\_wrapper\_hw\_platform\_0\system.mssTarget Processor: ps7\_cortexa9\_0

Operating System

Board Support Package OS.

Name: standaloneVersion: 6.2Description: Standalone is a simple, low-level software layer. It provides access to basic I/O and features of a hosted environment, such as standard input and output, profiling, and so on. It is not intended for use in a production environment or features such as real-time and exit.Documentation: [standalone v6 2](#)

Peripheral Drivers

Drivers present in the Board Support Package.

ps7\_afi\_0 genericDocumentationps7\_afi\_1 genericDocumentationps7\_afi\_2 genericDocumentationps7\_afi\_3 genericDocumentationps7\_coresight\_comp\_0 coresightps7\_dcc Documentationps7\_ddr\_0 ddrpsDocumentationps7\_ddrc\_0 genericDocumentationps7\_dev\_cfg\_0 devcfgDocumentationImport Examplesps7\_dma\_ns dmapsDocumentationImport Examplesps7\_dma\_s dmapsDocumentationImport Examples

OverviewSource

ProblemsTasksConsolePropertiesSDK Terminal

Program Flash

Performing Erase Operation...Erase Operation successful.INFO: [Xicom 50-44] Elapsed time = 10 sec.Performing Program Operation...0%...70%...80%...90%...100%Program Operation successful.INFO: [Xicom 50-44] Elapsed time = 79 sec.Flash Operation Successful


Target Connections

Hardware ServerLinux TCF AgentQEMU TcfGdbClient

将开发板打到QSPI\_Flash模式JP1闭合、JP2闭合，可以看到开发板三色灯D34不停的闪烁，如果将开发板电源断开，然后重新连接上可以看到开发板依然可以正常运行，可以看到开发板断电程序没有丢失







虚无缥缈vs威武

发布了37 篇原创文章 · 获赞 58 · 访问量 15万+



CorelDRAW Graphics Suite 2020

為您帶來市面上最快速、流暢、精準的向量圖形設計軟體。

广告 coreldraw.com



想对作者说点什么



jd\_1234567

8个月前

请问博主，例化那一步是怎么做的啊？能不能具体讲一下？

👍 2

🔗

💬 5

☆

📱

<

>

👤 赏

关注

回复(4)

👍

MYIR-ZYNQ7000系列-zturn教程(2)：Hello\_World

阅读数 3875

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1（工程末尾提供了工程源代码大家可以去网盘下载）ste... 博文 来自： taowei1314520的...

ZYNQ系列通过QSPI启动的程序烧写

阅读数 841

ZYNQ系列通过QSPI启动的程序烧写近来在调试ZYNQ与上位机的PCIe通信，因为上位机机箱重启BIOS检测方式设... 博文 来自： weixin\_44010639...

我花了一夜用数据结构给女朋友写个H5走迷宫游戏

阅读数 42万+

起因又到深夜了，我按照以往在csdn和公众号写着数据结构！这占用了我大量的时间！我的超越妹妹严重缺乏陪伴而... 博文 来自： bigsai

03-ZYNQ学习（启动篇）之程序的固化

阅读数 6540

上一节我们只使用ZYNQ的PL端资源，单独控制PL端的LED实现流水灯：https://blog.csdn.net/fengyuwuzu0519/... 博文 来自： 风雨无阻

ZYNQ 动态更新bit流 ZynqMP PL Programming

阅读数 74

项目最后的几个事宜，本以为很简单，结果深坑不浅。公司为某部分设计的X X板，设计1 2 8 M B QSPI Flash，... 博文 来自： maybe\_only的博客

【设计经验】3、ISE中烧录QSPI Flash以及配置mcs文件的加载速度与传输位宽

阅读数 59

一、软件与硬件平台 软件平台： 操作系统：Windows 7 64-bit 开发套件：ISE14.7 硬件平台：FPGA型号：XC6SL... 博文 来自： weixin\_30478923...

学会Zynq（2）Zynq-7000处理器的配置详解

阅读数 2788

上篇中介绍了Xilinx FPGA嵌入式开发的基本概念和软件特性，并以Hello World为例给出了一个操作流程，熟悉该流... 博文 来自： FPGADesigner的...

VIVADO2017.4无法下载QSPI解决方案

阅读数 1131

QSPI FLASH 下载，分析说明。从官方的文档中，经过分析和实际操作，总结以下信息。（1）“指定fsbl”文件修改... 博文 来自： Huskar\_Liu的博客

zynq 文件系统中加载PL fpga.bit笔记

阅读数 1368

1.在启动脚本中添加创建/dev/xdevcfg节点：if [ ! -e /dev/xdevcfg ] then mknod /dev/xdevcfg c 259 0fi2.直... 博文 来自： xld123

zynq7000只使用PL的程序打包与加载

阅读数 534

1、创建FPGA工程，设计PL程序并验证；2、Create Block Design，添加Zynq处理器，设置中把所有的外部IO删除... 博文 来自： rong81117的博客

MYIR-ZYNQ7000系列-zturn教程(17)：用axi\_uart发送数据

阅读数 4160

开发板环境：vivado 2017.1，开发板型号xc7z020clg400-1，这个工程主要用axi\_uart发送数据，IP核设置的波特... 博文 来自： taowei1314520的...



pang\_bo

2篇文章

关注 排名:千里之外



Big sai

199篇文章

关注 排名:2000+



【星星之火】

199篇文章

关注 排名:6000+

VIVADO 2017.4烧写QSPI FLASH

阅读数 1880

开发ZYNQ时，在VIVADO 2017.4在烧写QSPI FLASH时必须指定FSBL文件，貌似是17.3后新增的特性，指定默认生... 博文 来自： zkf0100007的博客

学会Zynq（6）固化程序到SD卡或QSPI Flash

阅读数 1765

SDK调试程序时都是通过JTAG，将PL的bit流文件和应用程序的ELF文件下载到Zynq中，运行查看效果。调试验证功... 博文 来自： FPGADesigner的...

zynq-7000的linux QSPI flash启动

阅读数 1万+


Xilinx SDK安装为了方便linux使用者，xilinx提供了一套SDK，使得linux BSP开发变得很轻松。第一步是下载Xilinx ... 博文 来自： gdlituo的博客





举报





<b>ZYNQ QSPI_FLASH程序固化</b>	阅读数 457
制作镜像文件需要：PL部分的bit文件，PS部分的elf文件，以及安置代码FSBL文件BOOT.bin = FSBL.elf+该工程.bit...	博文 来自： <a href="#">yaoyaoshalou的博...</a>
<b>请问vivado xc7z020引脚问题</b>	
xc7z020只有123个IO，我要用来处理192位的数据，请问怎么分配引脚呢，听说一位数据只能接一个IO。	论坛
<b>ZYNQ QSPI flash 启动完后，挂载的问题</b>	
QSPI 总大小16MB， 最后一个分区大小为5M，就是这个分区，我是用来做user space的， 但是想要挂载这个分区，一定...	论坛
<b>vivado烧写bin文件到flash 中</b>	阅读数 9168
点击 bitstream setting ，将 bin_file 勾上，点击 OK。 2) 点击 generate bitstream ，生成 bit 文件和 bin 文件...	博文 来自： <a href="#">坚持</a>
<b>Zynq7020 有关于vivado2017.4烧写qspi报错的解决办法</b>	阅读数 2579
参考：https://www.xilinx.com/support/answers/70148.html***** Xilinx Program Flash***** Program Flash ...	博文 来自： <a href="#">smile_5me的博客</a>
<b>SPIFI(QSPI)使用</b>	阅读数 5827
1. SPIFI 标准 SPIFI (SPI FLASH INTERFACE) ， 百度百科的定义：SPIFI是SPI闪存接口专利技术的缩写，可以...	博文 来自： <a href="#">CODING</a>
<b>vivado烧写flash</b>	阅读数 1万+
1) 点击 bitstream setting ，将 bin_file 勾上，点击 OK。 2) 点击 generate bitstream ，生成 bit 文件和 bin ...	博文 来自： <a href="#">u010830004的专栏</a>
<b>(电工基地笔记) Vivado固化至SPI Flash</b>	阅读数 1万+
如果从头开始做SPI Flash固化是有一些麻烦的，要在完成综合之后，打开synthesized Design（图）（图） 然后在...	博文 来自： <a href="#">Peter's Blog</a>
<b>ZYNQ中的QSPI FLASH烧写问题</b>	阅读数 4979
BSP生成问题使用SDK进行QSPI FLASH烧写时，应注意FSBL的生成问题，建立FSBL时不能选择新建BSP，而应选择...	博文 来自： <a href="#">fengmaoqiao的博客</a>
<b>vivado mcs文件的QSPI Flash固化</b>	阅读数 53
开发板的QSPI Flash连接到FPGA,用于fpga的比特流固化。若将fpga运行的比特流固化到QSPI Flash中。fpga每次上...	博文 来自： <a href="#">think ofu</a>
<b>如何在Zynq-7000上烧写PL Image</b>	阅读数 3188
在Zynq-7000上编程PL大致有3种方法：1. 用FSBL，将bitstream集成到boot.bin中2. 用U-BOOT命令3. 在Linux下...	博文 来自： <a href="#">菜头</a>
<b>vivado bit 烧写到flash</b>	阅读数 869
转自：https://blog.csdn.net/renlongggg/article/details/75127139原文地址：http://blog.sina.com.cn/s/blog_c...	博文 来自： <a href="#">yundanfengqing_...</a>
<b>Xilinx_Zynq_QSPI启动的编译、烧写、及配置过程步骤</b>	阅读数 1万+
Xilinx_Zynq_QSPI启动的编译、烧写、及配置过程步骤 一、 交叉编译器：使用xilinx提供的编译器arm-xilinx-linux-...	博文 来自： <a href="#">ZLM的博客 --- &gt;...</a>
<b>Vivado SPI Flash程序下载</b>	阅读数 7423
由于Vivado下载程序步骤和ISE有较大差异，特此写此文章，希望对大家有所帮助。1，下载文件生成在.bit文件生成...	博文 来自： <a href="#">家穷人瘦</a>
<b>zynq 7000下读写qspi 及flash 唯一id</b>	阅读数 6330
本文主要演示zynq 7000下对Qspi 的读写操作，以及读取8字节唯一ID, 可以用于简单加密。我在使用altera 的时候...	博文 来自： <a href="#">曾立文的博客</a>
<b>Zynq-Linux移植学习笔记之19-启动加载与固化</b>	阅读数 2145
1、 启动加载启动时候自动执行shell脚本Mount ramdisk，修改rootfs里面rcs文件 在RCS中添加代码echo "+ + Sta...	博文 来自： <a href="#">无知的我</a>
<b>ZYNQ_QSPI_FLASH烧写教程</b>	阅读数 4765
注意：烧写之前先保证程序在JTAG方式在线仿真没有问题。1. 在BLOCK上的ZYNQsystem的MIOConfigure添加...	博文 来自： <a href="#">weixin_37728585...</a>
<b>zedboard烧写程序到FLASH，用于QSPI Flash启动</b>	阅读数 1万+
创建第一级启动引导程序FSBL。SDK中点击工程BSP文件夹右键，选择Board Support Package Setting。选择xilff...	博文 来自： <a href="#">YunLan天地间</a>
<b>zynq qspi 增加jtag调试和固件升级设置，启动镜像。</b>	阅读数 1340
最近这几天想固件升级的问题，开始考虑双u-boot，然后在fsbl里面增加一个多镜像的操作，增加一段代码驱动网络...	博文 来自： <a href="#">u011529140的博客</a>
<b>爬虫福利二 之 妹子图网MM批量下载</b>	阅读数 27万+
爬虫福利一：27报网MM批量下载 点击看了本文，相信大家对爬虫一定会产生强烈的兴趣，激励自己去学习爬虫， ...	博文 来自： <a href="#">Nick.Peng 的博客</a>
<b>Java学习的正确打开方式</b>	阅读数 40万+
在博主认为，对于入门级学习java的最佳学习方法莫过于视频+博客+书籍+总结，前三者博主将淋漓尽致地挥毫于这...	博文 来自： <a href="#">程序员宜春的博客</a>
<b>程序员必须掌握的核心算法有哪些？</b>	阅读数 53万+
由于我之前一直强调数据结构以及算法学习的重要性，所以就有一些读者经常问我，数据结构与算法应该要学习到哪...	博文 来自： <a href="#">帅地</a>


2





5











04-29

09-02



举报



2019年4月	1篇
2019年3月	2篇
2019年1月	1篇
2018年11月	1篇

展开

热门文章

- [VIVADO 安装教程](#)  
阅读数 84216
- [三态门详解](#)  
阅读数 15398
- [quartus II 12.1 使用教程（1） 怎样调用PLL 核](#)  
阅读数 7556
- [MYIR-ZYNQ7000系列-zturn教程\(17\)：用axi\\_uart发送数据](#)  
阅读数 4156
- [MYIR-ZYNQ7000系列-zturn教程\(9\)：将bit文件固化到QSPI\\_Flash](#)  
阅读数 4055

最新评论

- [VIVADO 安装教程](#)  
rq8866：缺License的小伙伴 链接：https://pan.baidu.com/s/11mjkpyERdUH3q5C\_TpfQxQ ...
- [FT232H如何使用jtag接口](#)  
taowei1314520：[reply]qq\_42662835[/reply]我是直接对eeprom里写数据进去的，数据我已经 ...
- [FT232H如何使用jtag接口](#)  
taowei1314520：[reply]sssshhhhhhhh[/reply]这个vivado有这个usb驱动也需要安装一下， ...
- [FT232H如何使用jtag接口](#)  
sssshhhhhhhh：你好，插上电脑以后显示 USB Serial Conventor （仅配置了USB和EEPROM这 ...
- [MYIR-ZYNQ7000系列-z...](#)  
kuyunge：SPI一次是通信一个字节码？



- QQ客服
- kefu@csdn.net
- 客服论坛
- 400-660-0108
- 工作时间 8:30-22:00

[关于我们](#) [招聘](#) [广告服务](#) [网站地图](#)

京ICP备19004658号 经营性网站备案信息  
 公安备案号 11010502030143  
©1999-2020 北京创新乐知网络技术有限公司  
网络110报警服务  
北京互联网违法和不良信息举报中心  
中国互联网举报中心 家长监护 版权申诉

2

5

举报