

# **PicoRV32 DEMO应用指南**

**(版本号: V1.0)**

深圳市紫光同创电子有限公司

版权所有 侵权必究

## 文档版本修订记录

[illegible]

## 名词术语解释

[illegible]

## 目录

1.	概述.....	1
1.1.	介绍.....	1
1.2.	主要功能.....	1
1.3.	设计信息.....	1
1.4.	PicoSoC资源使用情况.....	2
2.	功能描述.....	7
2.1.	PicoSoC设计架构.....	7
2.2.	接口列表.....	8
2.3.	参数定义.....	8
2.4.	接口时序.....	7
3.	参考设计.....	7
3.1.	参考功能设计.....	7
3.2.	PicoSoC系统与配置.....	7
3.3.	参考设计文件目录.....	8
3.4.	参考设计仿真.....	8
3.5.	参考设计上板验证.....	8

## 图目录

图 1 PiCoSoC功能框图 .....	7
图 2 NMI外部读入指令典型时序 .....	7
图 3 NMI内部写出数据典型时序 .....	7
图 4 PiCoSoC配置情况图 .....	7
图 5 文件目录 .....	8
图 6 GPIO-OUT-LED仿真 .....	8
图 7 GPIO-IN-DATA仿真 .....	8
图 8 UART分频仿真 .....	8
图 9 UART打印仿真 .....	8
图 10 串口打印结果 .....	8

## 表目录

表 1 PiCoRV32软核信息 .....	1
表 2 资源使用率 .....	2
表 3 PiCoSoC接口列表 .....	4
表 4 PiCoRV32参数定义 .....	4
表 5 寻址空间总体划分表 .....	7

## 1. 概述

### 1.1. 介绍

本文档为深圳市紫光同创电子有限公司PicoRV32软核产品及应用文档。本文当主要介绍了PicoRV32软核的功能列表、设计架构、接口定义、接口时序、支持器件以及参考设计等。

PicoRV32软核是建立在精简指令集（Reduced Instruction Set Computing V, RISC-V）基础上的可扩展处理器，支持整数指令集（Integer, I）、乘除法指令集（Multiplication and Division, M）以及压缩指令集（Compact, C），其主要应用场景为嵌入式工控领域。

### 1.2. 主要功能

PicoRV32所组成的单核片上系统（PicoSoC）支持的主要功能如下：

- PicoSoC主频12-150MHz
- 软核资源占用3.7K LUTs
- 支持RV32IMC指令集
- 单指令平均周期数为4
- 冯诺依曼结构
- RAM指令初始化
- 支持UART与GPIO
- 支持本地存储接口（Native Memory Interface, NMI）

本参考设计是利用NMI总线对PicoRV32软核本身进行访问和操作，同时将UART与GPIO外设挂载在该总线上。根据C代码的功能设计，成功实现了UART数据打印、GPIO-OUT-LED闪烁，以及GPIO-IN数据输入三项功能，并验证了IMC指令集的兼容性。

### 1.3. 设计信息

表 1 PicoRV32软核信息

PicoRV32软核	
支持器件	PGT/PGL系列FPGA产品
支持用户接口	本地存储接口(NMI)
提供的设计文件	
PicoRV32设计文件	Verilog文件
PicoRV32参考设计	Verilog文件
PicoRV32仿真文件	Verilog文件
PicoRV32约束文件	fdc文件
开发工具支持	

RTL设计工具	PDS开发套件 Pango Design Suite 2019.1-patch2版本
软件设计工具	Windows x64版本的Eclipse开发工具 jdk-8u101-windows-x64 python3x64
仿真工具	ModelSim10.2c

## 1.4. PicoSoC 资源使用情况

表 2 资源使用率

器件	Distributed RAM	DRM	FF	LUT	PLL
PGL22G	32	16	1326	4089	1

## 2. 功能描述

目前，PicoRV32软核的主要场景是FPGA芯片外设配置和复杂状态逻辑实现。通过PicoRV32软核和相关配置组成PicoSoC，并在PGL22关键特性板卡上实现UART打印和简易GPIO功能。

### 2.1. PicoSoC 设计架构

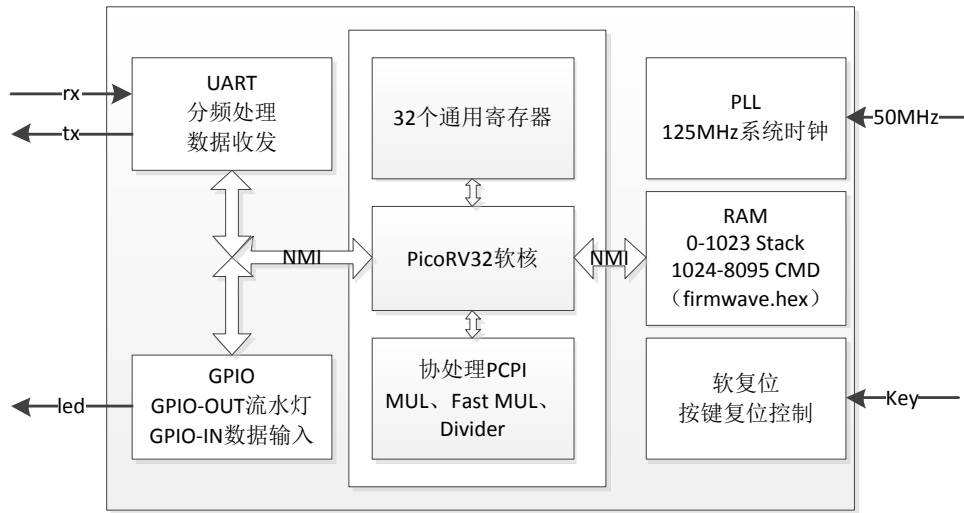


图 1 PicoSoC功能框图

#### PicoRV32软核模块：

该模块由PicoRV32软核，32个通用寄存器以及可选的协处理器组成。

协处理器包括乘法器（MUL）、快速乘法器（Fast MUL）以及除法器（Divider），并通过Pico Co-Processor Interface (PCPI)总线与PicoRV32软核连接，其中乘法器和快速乘法器不能同时使用。

协处理器部分与自定义的PCPI总线和中断处理等高级应用有关，详细介绍请参见README.md介绍，PicoSoC并未使用此项功能，因此不做介绍。

#### UART模块：

该模块由分频系数和数据收发控制组成，主要用于C代码调试中的字符打印和命令等外部数据的输入，其逻辑部分完全遵循UART协议，在此不对UART原理赘述。

该模块可配合C代码实现字符/字符串、十进制/十六进制/浮点数打印。

#### GPIO模块：

该模块是配合C代码，在固定地址处输出LED灯闪烁信息，同时将GPO输入数据写在另一固定地址处，软件端发出扫描申请，提出该地址处的数据。

#### 其他模块：

PLL、RAM、软复位/按键复位、较为普通不做介绍。



## 2.2. 接口列表

表 3 PicoSoC接口列表

信号名	I/O	位宽	描述
顶层信号			
clk_50m	I	1	晶振50MHz
resetrn	I	1	按键复位，低有效
ser_rx	I	1	UART数据接收
ser_tx	O	1	UART数据发送
led	O	8	8bit用户led灯
irq_5	I	1	保留
irq_6	I	1	保留
irq_7	I	1	保留
Native Memory Interface (NMI-本地存储接口)			
mem_valid	O	1	存储有效信号，高有效
mem_instr	O	1	指令指示信号 1: 指令 0: 数据
mem_ready	I	1	指令输入准备信号，高有效
mem_addr	O	32	PocoRV32软核地址输出
mem_wdata	O	32	PocoRV32软核数据输出
mem_wstrb	O	4	0000: 32位数据无效 1111: 32位数据无效 1100: 高16位数据有效 0011: 低16位数据有效 1000, 0100, 0010, 0001: 8位数据有效
mem_rdata	I	32	PocoRV32软核指令输入

## 2.3. 参数定义

表 4 PicoRV32参数定义

PARAMETER	描述	默认值
STACKADDR	堆栈大小，即栈底位置	32'h FFFF_FFFF
PROGADDR_RESET	首次地址首次输出值	32'h 0000_0000
PROGADDR_IRQ	中断编译地址	32'h 0000_0010
BARREL_SHIFTER	筒形移位寄存器	1'b0
COMPRESSED_ISA	压缩指令集使能	1'b0
ENABLE_MUL	乘法器使能	1'b0
ENABLE_DIV	触发器使能	1'b0
ENABLE_IRQ	中断使能	1'b0
ENABLE_IRQ_QREGS	中断 q0-q3 使能	1'b1
其余参数为默认值	详细请见 README.md	略

## 2.4. 接口时序

此处只介绍 NMI 接口时序,UART 等接口时序较为普通，不做赘述。

NMI 接口只有 mem\_ready 与 mem\_rdata 为输入，即可受 RTL 逻辑控制，其余接口均为 PicoRV32 软核内部输出，作为外部逻辑的控制信号或数据来源，因此输入信号是整个时序的关键。

其中，用户最为关心的是机器码指令读入时序（外部-“读出”、软核-“写入”）和软核内部数据写出时序（软核-“读出”、外部-“写入”），其典型时序如图2和图3所示。

值得注意的是，PicoRV32软核的单指令平均周期数为4，因此执行单个指令需要大约四个系统时钟（不是每个指令都需要四个时钟）。

不论读入还是写出时序，都要在mem\_valid拉高后，且mem\_ready高有效时才能进行操作。

mem\_instr为高时，代表此时输入软核的数据为可执行的机器码指令；为低时，代表此时可输出指令执行后的结果，与mem\_wdata有关。

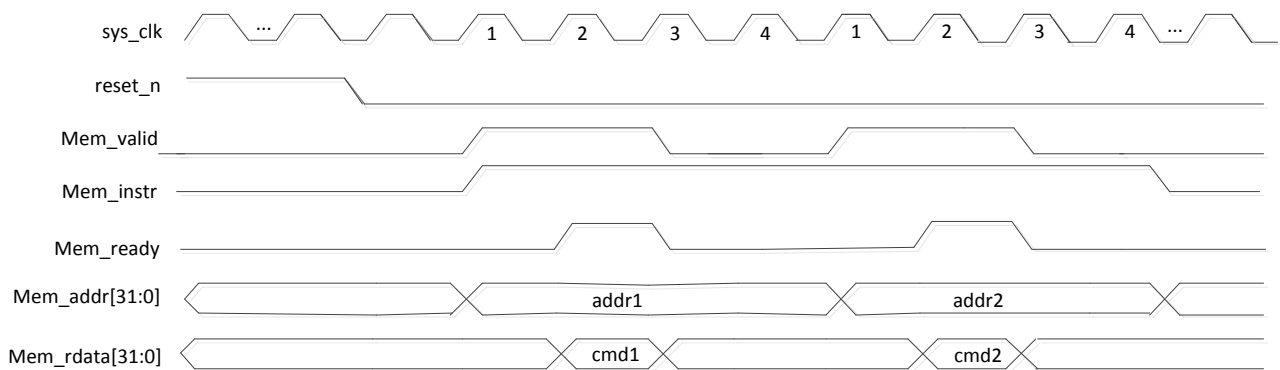


图 2 NMI外部读入指令典型时序

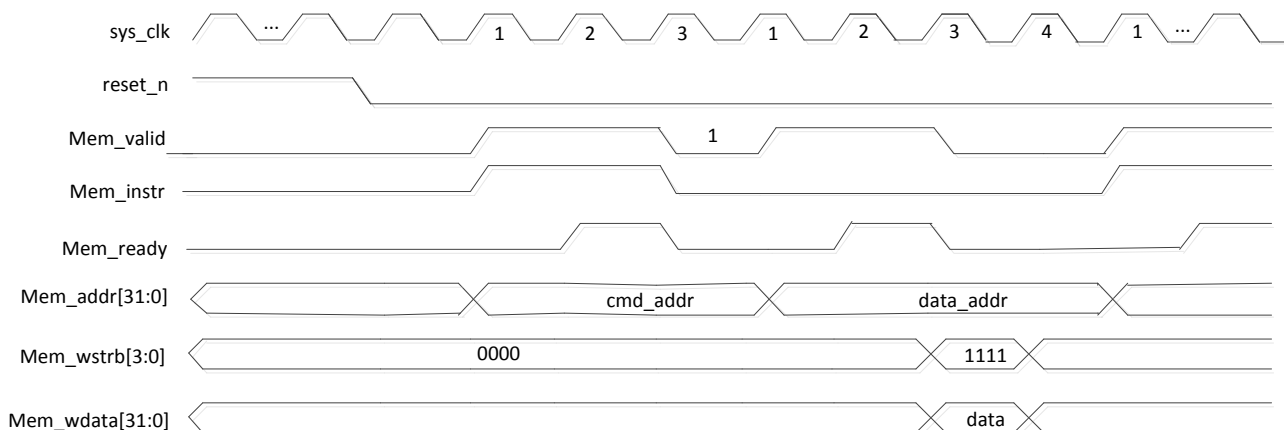


图 3 NMI 内部写出数据典型时序

## 3. 参考设计

### 3.1. 参考功能设计

PicoSoC功能设计如图1所示，系统时钟为125Mhz，内部设置软复位功能，其功能模块介绍请参见第二章-PicoSoC设计架构所述。此外，为符合本参考设计所要求的软件代码量和功能需求，设置堆栈大小为32bit\*1024，系统起始地址为32'h1000，详细配置如3.2节所示。

本章将介绍与软件端匹配的PicoSoC系统搭建与配置情况，而Eclipse软件开发工具使用方法和可执行的机器码指令（ram.hex）生成方法，详细请参见《PicoRV32 IDE Quick Start应用指南》。

### 3.2. PicoSoC 系统与配置

PicoSoC系统包括PicoRV32软核、UART、GPIO、RAM、PLL以及复位信号、LED灯等必要的信号，将外设直接挂载在NMI总线上，其读写时序请参见图2和图3所示。

将系统实例化调用，其配置如图4所示：

```
picorv32 #(
    .STACKADDR(4096),
    .PROGADDR_RESET(32'h0000_1000),
    .PROGADDR_IRQ(32'h0000_0000),
    .BARREL_SHIFTER(1),
    .COMPRESSED_ISA(1),
    .ENABLE_MUL(1),
    .ENABLE_DIV(1),
    .ENABLE_IRQ(1),
    .ENABLE_IRQ_QREGS(0)
) cpu (
    .clk          (sys_clk    ),
    .resetn       (reset_n   ),
    .mem_valid    (mem_valid  ),
    .mem_instr    (mem_instr  ),
    .mem_ready    (mem_ready  ),
    .mem_addr     (mem_addr   ),
    .mem_wdata    (mem_wdata  ),
    .mem_wstrb    (mem_wstrb  ),
    .mem_rdata    (mem_rdata  ),
    .irq          (irq       )
);
```

图 4 PicoSoC配置情况图

由图4可知，整个系统的堆栈栈顶地址为4096，软核起始地位为32'h1000，支持筒形移位、乘除法器、中断以及IMC指令集。

为配合软件端实现软核及外设的正常运行，就需要将32位的寻址空间进行划分，本参考设计是在FPGA中开辟一块大小为256Kb（32bit\*8192），且可读写的RAM空间。将堆栈划归在RAM的0到1023处，即32'h0-32'hFFc；机器码指令存储在1024到8191处，即32'h1000-32'h7FFC；UART分频配置地址为32'h8000，UART数据打印地址32'h8010；GPIO输出地址为32'h8020，GPIO输入地址为32'h8030，寻址空间总体划分如表5所示：

表 5 寻址空间总体划分表

地址范围	描述
32'h0000-32'h0FFC	堆栈

32'h1000-32'h7FFC	指令空间
32'h8000-32'h800C	UART分频
32'h8010-32'h801C	UART打印
32'h8020-32'h802C	GPIO输出
32'h8030-32'h803C	GPUO输入

此外,生成可被执行的ram.hex文件需要依赖初始化汇编代码-start.s、链接文件-section.lds、主程序及相关C代码和头文件-main.c,同时通过Makefile调用Eclipse开发工具中的GCC等工具将该操作完成,详情请参见《PicoRV32 IDE Quick Start应用指南》。

### 3.3. 参考设计文件目录

pgr\_PICORV32设计实例目录结构图:

bench	//仿真test bench
docs	//设计文档、PicoRV32 README
ip	//设计调用相关IP
pnr	//工程目录
generate_bitstream	// .sbit
ipcore	// IP中用到的IP
source	// fdc文件
picorv32.pds	// 工程文件
ram.hex	// 机器码hex文件
simulation	// 仿真工程目录
src	// 设计实例包含的RTL文件
picosoc	// soc代码
picorv32.v	// 软核本身
software	// Eclipse, JAVA, Python软件压缩包

图 5 文件目录

### 3.4. 参考设计仿真

由于关键的机器码和软核代码已经加载至RTL工程中,因此只需要提供时钟激励就可以开始仿真,以下依次是GPIO-OUT-LED仿真、GPIO-IN-DATA仿真(32'haa55)、UART分频仿真、UART打印仿真,各自时序都符合预计设计且完全正确。

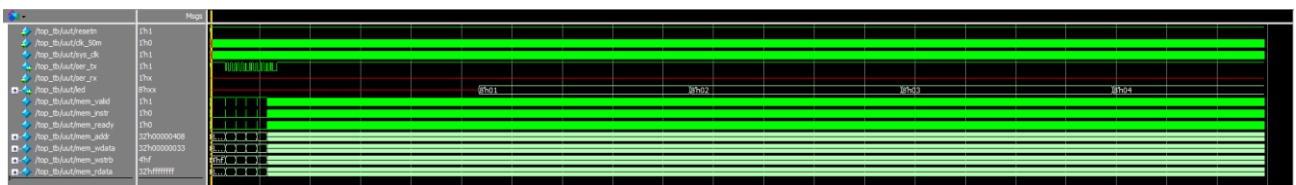
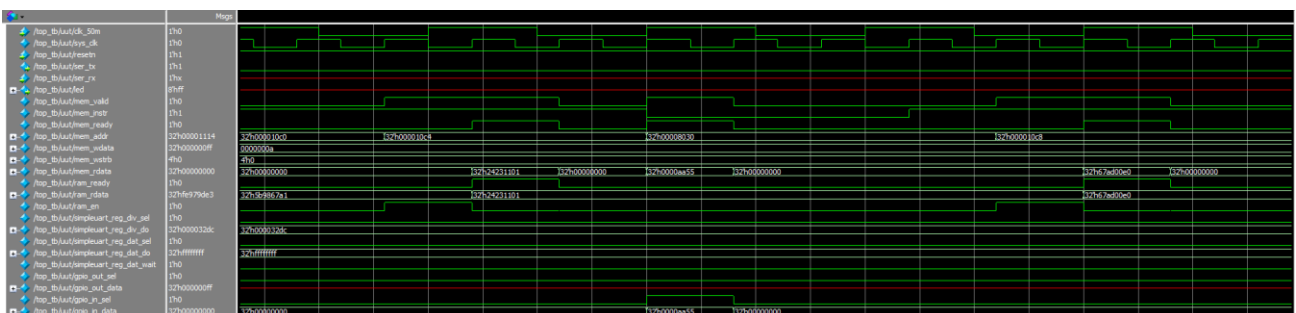


图 6 GPIO-OUT-LED 仿真



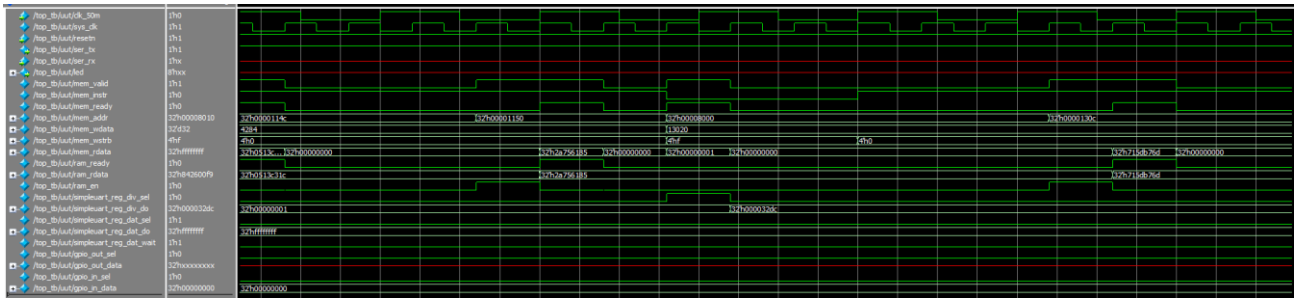


图 8 UART 分频仿真



图 9 UART打印仿真

### 3.5. 参考设计上板验证

- 1) 上板前确保工程代码正确，连接UART和约束管脚
- 2) 上电后，观察流水灯和串口输出结果，以及实际波形

LED闪烁点亮，串口成功打印“Hello Risc-V Pango 2019”字符串等信息，具体情况如图9所示：



图 10 串口打印结果