zCore 用户手册

版本	2.1
日期	2014.10.23
维护	lokisz@yahoo.com
发布范围	公开
适用范围	硬件和软件开发者

日期	版本	变更日志
2011.5.4	1.0	初始版本
2011.5.10	1.1	更新 uartlite 示例和 STA 寄存器使用注意事项；增加指令模拟部分说明；增加 PSS 的存储空间分配说明；
2011.5.16	1.2	更新 ocm 对程序的限制说明；zCore 增加 dcr 支持，确定 uic 和 uartlite 的 dcr 地址分配。修改 uartlite 示例。
2011.5.17	1.3	硬件更新 ocm 设计，统一指令和数据 ocm，支持读写和执行。
2011.9.29	2.0	根据产品计划修改 EDK 等说明，去除 UIC 部分说明，增加实现指南部分。
2014.10.23	2.1	更新维护邮件和增加开源邮件组

1，简介

zCore 处理器原为 PWRSemi 针对嵌入式市场开发的处理器系列 IP 核，目前以 GPLv2 许可证开源(https://groups.yahoo.com/neo/groups/openzcore/info)，zCore 处理器透明兼容 PowerPC/Power 架构指令集。本手册主要内容为介绍如何进行 zCore 的系统软件和应用开发。

第二部分将主要描述 zCore 架构和编程注意事项，软件开发者需依据 zCore 的不同实现，进行初始化，并相应配置编译器参数，以避免产生未实现指令并优化程序性能。

第三部分将主要描述 zCore 处理器子系统（PSS）的硬件系统架构，PSS 为一个最简单的完整系统，包括处理器内核，片上存储器，定时器，中断控制器和 UART 通讯模块等部分。

第四部分将介绍集成 PSS，片上总线和其他外围模块 IP 的参考 SoC 系统。

第五部分将描述入门级 EDK 平台－XUPv2P 硬件平台的具体使用，包括板上资源和各自功能定义，例如 LED 灯和开关等。其中板载的 FPGA 芯片用于实现参考 SoC 系统。

2, zCore 架构和编程

zCore 架构遵从“The IBM PowerPC Embedded Environment: Architectural Specifications for IBM PowerPC Embedded Controllers”规范[1]，具体实现参考 IBM ppc405 处理器核用户手册[2]。

根据实现特点和 Power 架构兼容性需要，zCore 将 ppc405 支持的全部指令集分为 I、II、III 三大类别。其中仅 I 类指令为实现 Power 架构兼容必须部分，II 类和 III 类指令在 zCore 处理器中皆为可选支持部分。完整 ppc405 指令集请参考[2]或附录 A PPC405 指令集。

2.1 指令集

2.1.1 I 类指令

I 类指令为 zCore 处理器提供基本 Power 兼容全部功能。这里的“基本 Power 兼容”含义是：提供“The IBM PowerPC Embedded Environment: Architectural Specifications for IBM PowerPC Embedded Controllers”中规定的用户级程序二进制兼容，即支持 Power 指令集嵌入式环境规范的 UISA 级全部指令，另外还包括部分 VEA 和 OEA 指令。I 类指令共有 191 条。

在开发过程中，为便于设计实现和验证，我们将 I 类指令进一步划分为 A 小类和 B 小类指令。

A 类指令采用硬件译码直接实现的方式。目前已实现全部 154 条 A 类指令

B 类，可配置 hard-wired、microcode、emulation，软件库或通过编译器关闭等方式实现兼容。包括

乘法指令 9 条－硬件实现

mulhw[.], mulhwu[.], mulli, mullw[o][.]

除法指令 8 条－未采用硬件方式实现－通过 emulation 实现

divw[o][.], divwu[o][.]

带更新的访存指令条 14 条－未实现，通过 GNU 编译器关闭

lbzu, lbzux; lhau, lhaux; lhzu, lhzux; lwzu, lwzux;

stbu, stbux; sthu, sthux; stwu, stwux;

复杂访存指令 6 条－未实现，通过 GNU 编译器关闭

lmw, lswi, lswx

stmw, stswi, stswx

部分硬件逻辑已经开发完成，但为进行集成和验证。

2.1.2 II 类指令

II 类指令为 Cache、MMU/TLB 相关指令。其中 4 条指令缓存指令、9 条数据缓存指令和 6 条 TLB 管理指令，共 19 条。未实现，为 NOP 操作。

2.1.3 III 类指令

III 类指令为 MAC、FPU、SIMD、SPE 和 VLE 等可选件的相关指令。其中 MAC 指令 84 条。未实现，引发 illegal 中断。

II 类和 III 类指令均为可选部分，并不影响 PowerPC 架构应用级的兼容性。

2.1.4 指令模拟

在目前的 zCore 中，对于硬件未支持的指令，设置特殊的 emualtion 空间来

模拟指令的执行。模拟 ROM 的空间暂定为 512B，地址暂定为（0xffff_fc00 - 0xffff_feff）。指令模拟 ROM 的地址可在硬件设计时配置（计划添加软件可写 SPR）。

目前 zCore 未支持的指令全部可通过指令模拟实现，包括通过编译器关闭的指令，以支持传统各 PowerPC 处理器的 legacy binary。同时，不同的模拟 ROM 配置可支持不同版本的 PowerPC 二进制文件。

指令模拟对系统软件和应用开发者透明，具体指令的执行周期可参考相关软件优化手册。

为提高指令模拟性能，zCore 包含多个非公开特殊 SPR 等。

2.2 寄存器组

zCore 寄存器组包括通用寄存器组 GPRs 和特殊寄存器组 SPRs。总共包括 32 个通用寄存器，条件寄存器 CR，机器状态寄存器 MSR 和其他通过 SPR 寻址的特殊寄存器，包括：

CTR 和 LR 寄存器；

定点例外寄存器 XER；

中断控制寄存器：DEAR, ESR, EVPR, MCSR, SRR0, SRR1, SRR2, SRR3；

通用 SPR：

USPRG0

SPRG0, SPRG1, SPRG2, SPRG3

SPRG4, SPRG5, SPRG6, SPRG7

定时器寄存器

TBL, TBU

PIT, TCR, TSR

除下列特殊说明部分寄存器，zCore 支持 ppc405 定义的用户态和超级用户态所有寄存器组和寄存器域定义的功能。

2.2.1 机器状态寄存器 MSR

ppc405 的 MSR 域可参考 ppc 用户手册[2]，目前 zCore 仅支持其中的四个位域，如下表所示：

位	位域名称	状态说明	备注
14	CE	Critical Interrupt Enable: 0 Critical interrupts are disabled 1 Critical interrupts are enabled	使能 critical 中断，watchdog timer 首次超时中断.
16	EE	External Interrupt Enable: 0 Asynchronous interruptsare disabled. 1 Asynchronous interrupts are enabled.	使能 non-critical external 中断, PIT, 和 FIT 中断.
17	PR	Problem State: 0 Supervisor state. 1 Problem state.	处理器状态设定
19	ME	Machine Check Enable 0 Machine check interrupts are disabled. 1 Machine check interrupts are enabled.	/

2.2.2 未实现的 SPRs

未实现的 SPRs 主要用于原 ppc405 的 mmu/cache 子系统和调试子系统，zCore 暂不支持上述功能。这些寄存器全部在 Privileged 模式下访问，包括有：

处理器版本寄存器：PVR（实际已经实现，但不建议软件使用）

用于 MMU 和 Cache 等的配置寄存器：CCR0, CCR1

存储控制寄存器：DCCR, DCWR, ICCR, SGR, SLER, SU0R

域保护寄存器：ZPR

调试支持寄存器：DAC1, DAC2, DBCR0, DBCR1, DBSR, DVC1, DVC2, IAC1, IAC2, IAC3, IAC4, ICDBDR

2.2.3 DCRs 的处理

PPC405 系统采用 CoreConnect 片上互连，部分外设寄存器映射为 DCR 寄存器，通过处理器的 mfdcr/mtdcr 访问。为适应不同片上总线，zCore 不推荐使用 DCR 配置外设寄存器。但为支持既有系统，目前 zCore 支持 mfdcr/mtdcr 指令，部分和处理器结合紧密的外部控制器，例如通用中断控制器 uic 和 uartlite，映射为 DCRs，为简化 zCore 内部逻辑，DCR 的地址限制为十六进制“x5x”，即第二位固定为 5。

2.3 中断处理

为方便处理器验证和系统调试，zCore 除实现 ppc405 的中断处理规范外，另外实现了特殊模式，可在芯片设计实现时配置。特殊模式和 ppc405 模式的区别仅在与中断向量地址的重定位和单个中断向量空间压缩，以减低存储器空间需求，适应最简系统设计。

2.3.1 PPC405 中断向量表－正常模式

中断类型	中断向量地址偏移	zCore 实现
Critical input interrupt	0x0100	支持
Machine check—data/instruction	0x0200	支持
Data storage interrupt	0x0300	暂未实现
Instruction storage interrupt	0x0400	暂未实现
External interrupt	0x0500	支持
Alignment	0x0600	支持
Program	0x0700	支持
FPU Unavailable	0x0800	暂未实现
System Call	0x0c00	支持
APU Unavailable	0x0f20	暂未实现
PIT	0x1000	支持
FIT	0x1010	支持
Watchdog timer	0x1020	支持
Data TLB miss	0x1100	暂未实现
Instruction TLB miss	0x1200	暂未实现
Debug	0x2000	暂未实现

2.3.2 特殊模式中断向量表

中断类型	中断向量地址偏移	zCore 实现
Critical input interrupt	0xff00	支持
Machine check—data/instruction	0xff10	支持
Data storage interrupt	0xff20	暂未实现
Instruction storage interrupt	0xff30	暂未实现
External interrupt	0xff40	支持
Alignment	0xff50	支持
Program	0xff60	支持
FPU Unavailable	0xff70	暂未实现
System Call	0xff80	支持
APU Unavailable	0xff90	暂未实现
PIT	0xffa0	支持
FIT	0xffb0	支持
Watchdog timer	0xffc0	支持
Data TLB miss	0xffd0	暂未实现
Instruction TLB miss	0xffe0	暂未实现
Debug	0xfff0	暂未实现

在系统初始化时，处理器的中断向量前缀寄存器 EVPR 配置为顶端地址 0xffff。这样可将中断处理程序定位于顶端存储空间（iocm）。外部通用中断控制器的地址偏移同样为硬件可配置参数，设置可参考后续章节。

另外，目前的 zCore 实现中，当发生非对齐的存储访问时，都会引发非对齐中断。可参考后续编程注意事项。

2.4 编程注意事项

对于 zCore 的系统软件和应用软件开发，须注意以下事项：

1，在针对目前的 zCore 200 进行高层语言编程时，须添加如下 gcc 参数：

"-mcpu=common"：以避免生成 4xx 系列专用的 mac 指令

"-mno-multiple"/"-mno-string"：关闭多字和 string 访存指令

"-mno-update"：关闭基址更新的访存指令

另外，zCore 暂时关闭了硬件除法器，以降低资源需求

2，对于系统软件开发者，关闭所有涉及 Cache、MMU/TLB 等的配置，并不使用相关指令，目前实现中，zCore 将 II 类指令视为空操作（或引发异常）。

3，zCore 支持 mfdcr/mtdcr 指令访问外设寄存器，但外设 DCR 的基址有所限制，具体可参考后续系统部分说明。

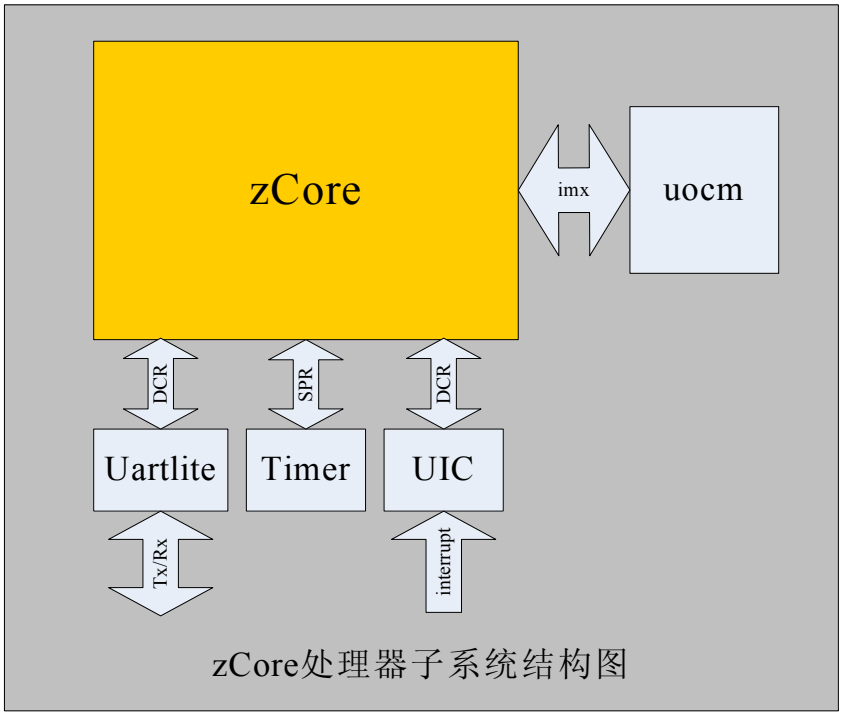
4，非对齐访问：所有未对齐的存储访问都会引发非对齐中断，不仅仅包括 ppc405 的原子指令对“lwarx/stwcx.”和 cache 指令等。这里未对齐的定义是指半字访问的末位地址为 1，字访问的未两位地址不是 00 等。

5，当对 zCore 不支持的 SPR 寄存器或未实现的寄存器域进行访问时，系统行为未定义。

2.5 软件开发工具

所有针对 PowerPC/Power 架构处理器的开发工具均可用于 zCore 的软件开发，例如 Codesourcery，DENX，Ronetix 和 Macraigo 等提供的基于 GNU 和 Eclipse 的开发套件。为保持一致性，推荐使用 Codesourcery 的系列工具链。

3, zCore 处理器子系统



zCore处理器子系统结构图

zCore 处理器子系统简称为 PSS(processor sub-system)，PSS 的结构如上图所示，外部 I/O 包括串口的收发接口，外部中断请求等。PSS 包含 zCore 处理器内核，统一的指令和数据片上存储器(uocm)，定时器 timer，可支持 32 个外部中断源的通用中断控制器 uic，以及用于调试的串口通信控制器 uartlite。timer 的寄存器访问通过 mfspr 和 mtspr 进行。uic/uartlite 的寄存器通过 mfdcr 和 mtdcr 完成。

3.1 片上存储器－OCM

片上存储器特性包括：

可配置大小：配置范围为取决于 FPGA 容量和 ASIC 工艺；

uocm 的访问对程序员透明－目前暂不支持可配置基址；

ucom 整个空间支持程序的“Write/Read/Execute”，但使用时建议限定空间用途，避免发生自修改代码情况（self-modify code）。

注意：对 uocm 进行非字（Byte 和 Halfword）的 store 访问时有一个周期的性能损失。

整个 uocm 的存储空间可通过芯片实现时初始化或者通过 uartlite 的特殊模式

实现在线改写。示例存储空间分配如下（配置为 16KB 时）：

* address: 0xffff_c000 - 0xffff_ffff.
* 0x0000_0000 - 0xffff_bfff: Non-Available
* 0xffff_c000 - 0xffff_d7ff: RAM, 7KB, "Read/Write", for program run-time data
* 0xffff_d800 - 0xffff_dfff: RAM, 1KB, "Read", for read only data
* 0xffff_e000 - 0xffff_ffff: 8KB, "Execute", for code section, includes:
  0xffff_e000 - 0xffff_fbff: user programs
  0xffff_fc00 - 0xffff_feff: emulation rom
  0xffff_ff00 - 0xffff_fff8: exception vector for special mode
  0xffff_fffc: reset vector

以上示例为硬件 SIM 和 FEP 环境验证汇编程序开发的实际环境，具体可参考配套示例文件。SIM 为 RTL 级验证环境，FEP 为 FPGA 仿真系统。

3.2 定时器－Timer

zCore 的定时器遵从 ppc405 的 timer 规范，可参考 ppc405 用户手册[2]

3.3 串口通信控制器－Uartlite

Uartlite 为 zCore 独有的串口通信模块，包括两种工作模式。正常模式下可用于和 pc 机通信，文件下载模式下可通过 pc 机端将 bin 格式的二进制启动文件下载到处理器子系统的 iocm 存储器。正常模式和下载模式的切换由平台硬件开关设定，具体使用方式可参考“系统板使用指南”。目前 uartlite 波特率固定为 115200，两个 stop 位，启动后无需软件配置。

Uartlite 正常模式的编程说明：包含 4 个 8 位寄存器，如下表，内部寄存器均通过 mtspr/mfspr 指令访问

DCR 地址	寄存器名字	位宽	访问	说明
0x350	TX	32 bit	写/读	发送数据寄存器：
0x351	RX	32 bit	只读	接收数据寄存器：
0x352	CTRL	32 bit	写/读	控制寄存器：
0x353	STA	32 bit	只读	状态寄存器：

3.3.1 发送数据寄存器 TX

位	0-23	24:31
位说明	保留	待发送数据

3.3.2 接收数据寄存器 RX

位	0-23	24:31
位说明	保留	接收数据

3.3.3 控制寄存器 CTRL

位	0-26	27	28	29	30	31
域说明	保留	中断请求清除	接收中断使能	发送中断使能	接收关闭	发送启动

注：1，复位状态为：32'h0000_0000

2，bit[31]/[27]写入“1”之后自动清零

3.3.4 状态寄存器 STA

位	0-26	27	28	29	30	31
域说明	保留	接收中断请求：1 表示接收完成一个字节	发送中断请求：1 表示前一个字节发送完成	中断请求：1 表示存在发送和接收完成的中断请求	接收状态：1 表示接收中，0 表示空闲	发送状态：1 表示忙，0 表示空闲

编程注意事项：STA 的使用，在读取 STA 状态之前，必须插入 3 条不影响 STA 状态的指令，以等待 STA 更新完成。

3.3.5 Uartlite 编程示例

Uartlite 编程示例 1－发送字符

```
addis r4, r0, ASCII_O@h
ori r4, r4, ASCII_O@l
mtdcr DCRN_DSUTX, r4      // data ready
mfdcr r5, DCRN_DSUSTA
andi. r5, r5, DSUTX_BUSY
bne (-8)                  // check transmitter status, if busy, waiting
addis r6, r0, DSUTX_GO@h
ori r6, r6, DSUTX_GO@l
mtdcr DCRN_DSUCTRL, r6    // send tx command to register dsu_ctrl
```

上述示例为发送 ASCII 码“O”（ASCII_O）工作流程：首先将要发送数据写入发送寄存器 TX，然后查询发送状态，如果忙则等待上一个发送完成，最后将发送命令写入控制寄存器 CTRL。注意，当 uartlite 正在发送中，接收到新的发送数据时，对当前发送不会产生影响。

Uartlite 编程示例 2－接收字符：

```
mfdcr r5, DCRN_DSUSTA
andi. r5, r5, DSURX_DONE
beq (-8)                  // check rx_done bit of STA; if no rx, wait
addis r7, r0, DSUIRT_CLEAR@h
ori r7, r7, DSUIRT_CLEAR@l
mtdcr DCRN_DSUCTRL, r7    // clear rx_done bit of STA via clear interrupt
```

上述示例为发送 ASCII 码“O”（ASCII_O）工作流程：首先查询发送状态，如果忙则等待上一个发送完成，然后将要发送数据写入发送寄存器 TX，将发送命令写入控制寄存器 CTRL。完整示例可参考汇编测试文件。

注意事项：1，目前 uartlite 工作在 polling 模式下，中断使能未使用。2，由

软件负责流控管理等，即如果处理器不对接收数据处理，下一个接收完成时将覆盖之前数据；接收完成一个字符之后，处理器必须清除接收完成标志，以便下次接收。

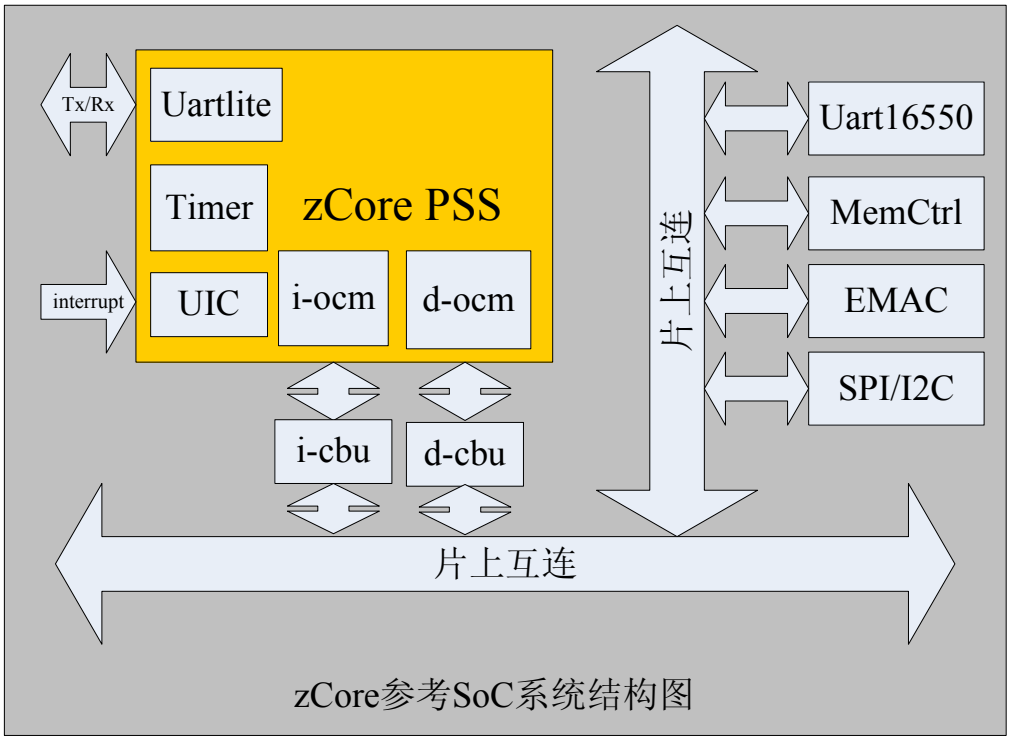
4, zCore 实现指南

经过验证的流程采用 Synplify 9.6.2 和 ISE 9.1-EDF 实现流程，仿真软件为 Modelsim 6.1f。

Pippo 的 PSS 系统初步实现结果如下表：

Design	Frequency	Resource	Device
pippo_core	92.6MHz	RegBits: 2369(8%); DPRams(RAM32X1D):96; BMUL:4; LUTs: 6857(23%)	Xilinx Virtex II Pro 30 Xc2vp30fg676-7
pippo_core	40.9MHz	RegBits: 1751 (7%); DPRams (RAM16X1D): 192; BRams: 2 of 28 (7%); LUTs: 7145 (32%)	Xilinx Sparton 3S xc3s1200eft256-4
top_pss (iocm/docm)	88.7MHz	RegBits: 2203(7%); DPRams(RAM32X1D):96; BRams: 72(52%); BMUL:4; LUTs: 6591(22%)	Xilinx Virtex II Pro 30 Xc2vp30fg676-7
top_pss(uocm)	75MHz	RegBits: 2510(8%); DPRams(RAM32X1D):96; BRams: 8(5%); BMUL:4; LUTs: 6891(23%)	Xilinx Virtex II Pro 30 Xc2vp30fg676-7

5, zCore 集成指南



参考 SoC 如上图所示，PSS 通过 CBU（Core Bridge Unit）支持各种片上互连协议，其他集成的外围模块包括 UART16550，存储控制器，以太 MAC 控制器和 SPI/I2C 等。

其他进一步规范待补充

5.1 内核桥接单元 CBU

目前 CBU 为完成集成和验证

5.2 参考 SoC

目前未完成 CBU 的 Wishbone 桥接和 AMBA 桥接。

6, FPGA 平台使用指南

XUPv2P 平台基于 Xilinx 的 XUPv2P 开发板[3]。目前已验证 zCore 的实际运行和 UART 文件下载等功能。详细试用可参考/board 目录源代码。

参考文档

[1] The IBM PowerPC Embedded Environment: Architectural Specifications for IBM

PowerPC Embedded Controllers, Version 1.0
[2] PPC405-S Embedded Processor Core User's Manual, Version 1.0
[3] XUP Virtex-II Pro Development System Hardware Reference Manual

附录 A：PPC405 指令集

按照功能分类，IBM ppc405 实现的指令集概括如下：

a.1 算术和逻辑运算指令

算术和逻辑运算指令包括 three-operand 格式和 two-operand 格式，以及立即数格式。[o]表示指令包含更新 XER[SO, OV]的“o”格式，和“non-o”格式；[.[表示指令包含更新 CR[CR0]的“record”格式，和“non-record”格式。其中有 24 条加法指令，加法指令 21 条，乘法指令 9 条，除法指令 8 条和 4 条负数指令，共 64 条算术运算指令，以及 28 条逻辑运算指令。注意：addic.，andi.和 andis.指令编码不包含 Rc 域[.]

算术运算指令如下表所示

Add	Subtract	Multiply	Divide	Negate
add[o][.] addc[o][.] adde[o][.] addi addic[.] addis addme[o][.] addze[o][.]	subf[o][.] subfc[o][.] subfe[o][.] subfic subfme[o][.] subfze[o][.]	mulhw[.] mulhwu[.] mulli mullw[o][.]	divw[o][.] divwu[o][.]	neg[o][.]

逻辑运算指令如下表所示

And	And with Complement	Nand	Or	Or with Complement	Nor	Xor	Eqv	Misc
and[.] andi. andis.	andc[.]	nand[.]	or[.] ori oris	orc[.]	nor[.]	xor[.] xori xoris	eqv[.]	extsb[.] extsh[.] cntlzw[.]

a.2 比较指令

比较指令用于对两个操作数进行算术或逻辑比较，并根据结果设置 CR。共有 4 条指令。

Arithmetic	Logical
cmp cmpi	cmpl cmpli

a.3 循环移位指令

共 14 条指令。

Rotate	Shift
rlwimi[.] rlwinm[.] rlwnm[.]	slw[.] sraw[.] srawi[.] srw[.]

a.4 CR 逻辑指令

共有 9 条指令

crand
crandc
creqv
crnand
crnor
cror
crorc
crxor
mcrf

a.5 分支指令

分支指令能无条件或者在一定条件下跳转到任意地址。条件分支指令首先测试先前指令设定的条件码，条件分支指令也能改变和测试 CTR 寄存器，也能将返回地址存入 LR 寄存器。条件分支的目标地址既可以来自当前地址或绝对地址的变换，也可以来自 LR 或 CTR。共有 4 条无条件跳转指令和 8 条条件跳转指令。

Unconditional	Conditional
	bc
	bca
b	bcl
ba	bcla
bl	bcctr
bla	bcctrl
	bclr
	bclrl

a.6 访存指令

访存指令用于在存储器和 GPR 之间传输数据，操作数包括字节 bytes，半字 halfwords 和字 words。另外，访存指令还支持 loading or storing multiple registers, character strings 和 byte-reversed data 等。共有 22 条 Load 指令和 18 条 Store 指令。

Loads					Stores			
Byte	Halfword Algebraic	Halfword	Multiple And String	Word	Byte	Halfword	Multiple And String	Word
lbz lbzu lbzux lbzx	lha lhau lhaux lhax	lhbrx lhz lhzu lhzux lhzx	lmw lswi lswx	lwarx lwbrx lwz lwzu lwzux lwzx	stb stbu stbux stbx	sth sthbrx sthu sthux sthx	stmw stswi stswx	stw stwbrx stwu stwux stwx stwcx.

a.7 中断控制指令

中断控制指令用于在 GPR 和 MSR 之间传输数据，从中断返回，使能或关闭 maskable external interrupts。共有 6 条指令。

mfmsr
mtmsr
rfi
rfic
wrtee
wrteei

a.8 处理器管理指令

处理器管理指令用于在 GPR 和 SPR 之间交换数据，提供 Traps，system calls 和同步控制。注意 DCR 相关指令的实现待定。共有 14 条指令。

eieio isync sync	mcrxr mfcr mfdcr mfspr mftb	mtcrf mtdcr mtspr sc tw twi

a.9 缓存控制指令（暂未支持）

缓存控制指令和以下的 TLB 管理指令暂未在 pippo 中实现。在 Pippo 的实现中，暂时按照 NOP 执行；共有 4 条指令缓存指令和 9 条数据缓存指令。

D-cache	I-Cache
dcba dcbf dcbi dcbst dcbt dcbtst dcbz dccci dcread	icbi icbt iccci icread

a.10 TLB 管理指令（暂未支持）

TLB 管理指令用于访问 MMU 中 TLB 的条项，搜索 TLB 阵列，失效 TLB 条项和同步其他处理器的 TLB 更新。共有 6 条指令。

tlbia tlbre tlbsx[.] tlbsync tlbwe

a.11 PPC405 特殊指令

注意，上述指令包含部分属于目标兼容产品 PowerPC405 实现，但并不包含在“IBM PowerPC Embedded Environment”规范中的指令。共 97 条指令，其中 84 条为 MAC 指令。

dccci dcread iccci icread	macchw[o][.] macchws[o][.] macchwsu[o][.] macchwu[o][.] machhw[o][.] machhws[o][.] machhwsu[o][.] machhwu[o][.] maclhw[o][.] maclhws[o][.]	nmacchw[o][.] nmacchws[o][.] nmachhw[o][.] nmachhws[o][.] nmaclhw[o][.] nmaclhws[o][.]	mulchw[.] mulchwu[.] mulhhw[.] mulhhwu[.] mullhw[.] mullhwu[.]	mfdcr mtdcr rfci tlbre tlbsx[.] tlbwe wrtee wrteei

	maclhwsu[o][.] maclhwu[o][.]			