



SUNWAY 申威

申威 1621 处理器 结构手册

2017 年 10 月

成都申威科技有限责任公司



免责声明

本档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

成都申威科技有限责任公司

Chengdu Sunway Technology Corporation Limited

地址：成都市华府大道四段电子科大科技园 D22 栋

Building D22, National University Science and technology park,
Section 4, Huafu Avenue, Chengdu

Mail: sales@swcpu.cn

Tel : 028-68769016

Fax: 028-68769019



阅读指南

《申威 1621 处理器结构手册》主要描述了申威 1621 处理器的核心结构、流水线组织、芯片结构、存储体系结构、终端异常说明、低功耗及可靠性设计等内容。

文档修订

文档更新记录	文档名	申威 1621 处理器结构手册
	版本号	V1.0
	创建人	研发部
	创建日期	2017-10-8

版本更新

版本号	更新内容	更新日期
V1.0	初稿	2017-10-8

技术支持

可通过邮箱或问题反馈网站向我司提交产品使用的问题，并获取技术支持。

售后服务邮箱：sales@swcpu.cn

问题反馈网址：<http://www.swcpu.cn/>

目 录

1	概述	1
1.1	核心	1
1.2	芯片	3
1.3	主要技术指标	5
2	核心结构	7
2.1	指令部件	7
2.1.1	转移预测器	9
2.1.2	指令流地址转换	9
2.1.3	取指控制	9
2.1.4	指令译码	9
2.1.5	寄存器重命名	10
2.1.6	指令发射	10
2.1.7	重排序缓冲	10
2.1.8	中断和异常管理	11
2.2	整数执行部件	11
2.2.1	整数寄存器文件	12
2.2.2	整数功能部件	12
2.3	浮点执行部件	14
2.3.1	浮点寄存器文件	16
2.3.2	浮点功能部件	16
2.4	数据Cache 控制部件	16
2.4.1	装入队列	18
2.4.2	存储队列	20
2.4.3	不命中地址文件	20
2.4.4	数据流地址转换缓冲	20
2.4.5	一级数据Cache 访问流水线	21
2.4.6	查询处理部件	21
2.5	二级Cache 控制部件	21
2.5.1	不命中地址文件	22
2.5.2	淘汰地址文件和淘汰数据缓冲	22
2.5.3	二级Cache 访问流水线	22
3	核心流水线组织	25
3.1	指令流水线	25
3.1.2	读指令Cache 标记站台	25
3.1.3	读指令Cache 数据站台	25
3.1.4	转移分析站台	25
3.1.5	指令整理站台	26
3.1.6	指令译码站台	26
3.1.7	寄存器重命名站台	26
3.1.8	指令预发射站台	26
3.1.9	指令发射站台	26
3.1.10	读寄存器站台	27
3.1.11	执行站台	27
3.1.12	指令完成站台	28
3.1.13	指令退出	28
3.2	运算流水线	28
3.3	访存流水线	29
4	芯片结构	30
4.1	片上互连结构	30

4.2	共享存储结构	30
4.3	一致性处理部件	30
4.3.1	三级 Cache	32
4.3.2	一致性处理控制	32
4.3.3	存储器访问缓冲	33
4.4	存储控制器	33
4.4.1	控制模块	33
4.4.2	数据通路	34
4.4.3	存储器接口	34
4.5	PCI-E 接口部件	35
4.6	集中控制部件	35
4.6.1	维护接口子部件	35
4.6.2	中断控制子部件	37
4.7	时钟接口	37
4.8	测试接口	37
5	存储体系结构	39
5.1	层次化存储体系结构	39
5.1.1	寄存器	40
5.1.2	指令 Cache	40
5.1.3	数据 Cache	40
5.1.4	二级 Cache	40
5.1.5	三级 Cache	42
5.1.6	主存	42
5.2	存储空间	42
5.2.1	虚空间	42
5.2.2	物理空间	44
5.3	存储管理	44
5.3.1	指令流地址转换缓冲	44
5.3.2	数据流地址转换缓冲	45
5.4	一致性处理机制	45
6	中断和异常	48
6.1	中断	48
6.1.1	中断类型	48
6.1.2	中断处理	50
6.2	异常	50
7	低功耗设计与管理	52
7.1	低功耗设计	52
7.1.1	门控时钟	52
7.1.2	多时钟设计	52
7.1.3	多阈值晶体管设计	52
7.1.4	运行控制	53
7.2	功耗管理	54
7.2.1	关闭不使用的核心、存储控制器	54
7.2.2	核心睡眠	54
7.2.3	部件切割	54
7.2.4	核心调速	54
8	可靠性设计	55
8.1	校验	55
8.2	容错	55
8.3	冗余	56
8.4	电源控制	56

1 概述

申威 1621 处理器采用多核结构和片上系统（SoC: System on Chip）技术，在单芯片中集成了 16 个同构的 64 位 RISC 结构的改进型第三代申威处理器核心（简称 Core3A），同时还集成了 32MB 的共享三级 Cache、8 路 64 位位 DDR3 存储控制器，以及第三代标准 PCI-E 接口和维护接口组成的系统接口。

1.1 核心

申威 1621 处理器包含 16 个同构的核心（Core3A），每个核心都是一个 64 位通用处理器。核心

的基本结构如图 1-1 所示。

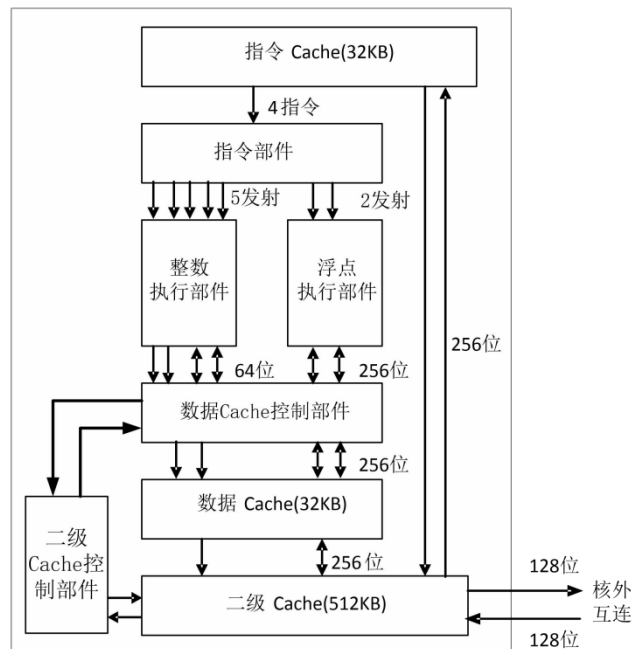


图 1-1: 申威 1621 处理器核心结构图

申威 1621 处理器核心由指令部件、整数执行部件、浮点执行部件、数据 Cache 控制部件、二级

Cache 控制部件以及一级指令 Cache、一级数据 Cache 和二级 Cache 组成。其特点有：

- 1) 核心为采用并行发射、乱序发射、乱序执行和推测执行技术的 4 译码 7 发射超标量结构；
- 2) 采用短向量加速计算技术提高整数和浮点运算性能，峰值运算速度为每个时钟周期可产生 11 个字整数运算结果或 16 个双精度浮点运算结果；
- 3) 指令 Cache 容量为 32KB，采用四路组相联结构，虚地址访问方式，Cache 行大小为 128

字节，采用可容错的偶校验；

4) 数据 Cache 容量为 32KB，采用四路组相联结构，物理地址访问方式，Cache 行大小为 128

字节，采用可纠错的 ECC 校验；

5) 二级 Cache 容量为 512KB，采用八路组相联结构，物理地址访问方式，Cache 行大小为 128

字节，为指令和数据混合 Cache，采用可纠错的 ECC 校验；

6) 数据 Cache 内容为二级 Cache 的子集，硬件支持 Cache 一致性；指令 Cache 内容既不是 SCache 的子集，也不与 SCache 互斥，可通过配置选择硬件支持指令 Cache 一致性或由软件解决一致性。

1.2 芯片

申威 1621 处理器芯片结构如图 1-2 所示：

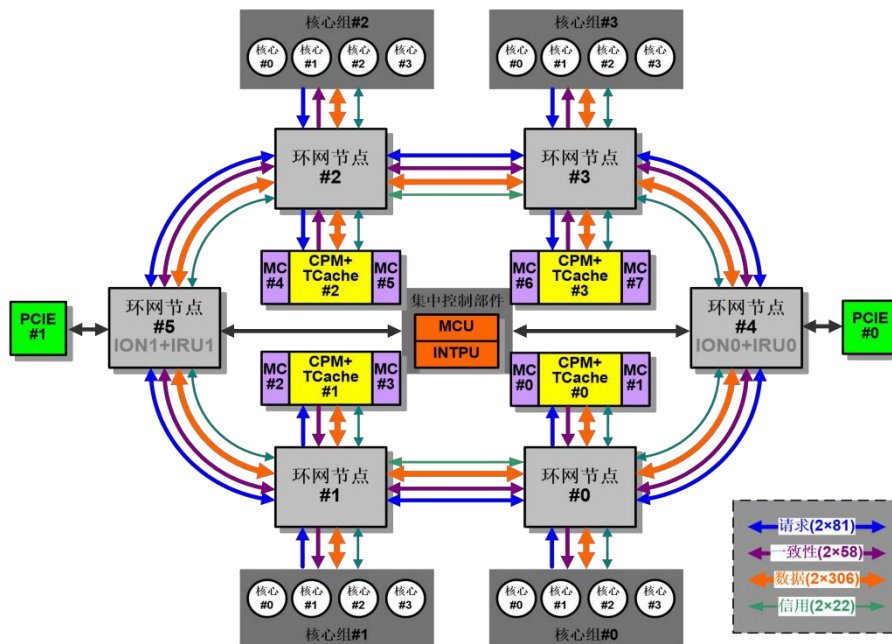


图 1- 2：申威 1621 处理器芯片结构图

申威 1621 处理器芯片采用片上环网互连 16 个核心、32MB 共享三级 Cache、8 路 64 位 DDR3

存储控制器、2 个 PCI-E 接口和维护接口组成的系统接口。其特点有：

- 1) 全芯片通过 4 套环网实现互连，环网上有 6 个节点。每 4 个核心构成一个核组，并与 8MB 的三级 Cache (TCache) 分体、一致性处理部件 (CPM)，以及两路 64 位 DDR3 存储控制器 (MC) 一起共享一个环网节点。2 个 8 链路的 PCI-E 接口分别连接一个环网节点。
- 2) 三级 Cache 分体的容量为 8MB，包括数据存储器 (TData) 和标记存储器 (TTag)。全

芯片 共包含 4 个三级 Cache 分体，总容量为 32MB。各核心 Cache 分布共享三级 Cache，并与三 级 Cache 间维持严格的副本包含关系（核心 Cache 总是 TCache 的子集）；

- 3) 一致性处理部件基于三级 Cache “标记存储器”的方法，实现核组内多 Cache 的一致性；
- 4) 全芯片共集成 8 路 DDR3 存储控制器，每路存储控制器通过 64 位数据通道连接片外的 DDR3 存储器，DDR3 存储器接口支持可纠错的 ECC 校验；
- 5) 2 个 8 链路的 PCI-E 的信号传输速率为 8Gbps，总的双向有效带宽为 32GB/s，支持各核心对 PCI-E 接口的访问，也支持 PCI-E 接口对三级 Cache 和主存的一致性访问；
- 6) 维护接口实现对申威 1621 处理器的状态监测、配置、初始化、自引导和维护控制，支持各核心对维护接口的访问，支持维护接口对存储器的一致性访问和直接访问；
- 7) 支持任意核心之间的中断传输，支持异步消息中断，支持维护接口和 PCI-E 接口对所有核心的中断；
- 8) 支持对核心工作频率、存储控制器工作频率和系统接口工作频率的独立控制；
- 9) 支持以核心为单位的运行管理和低功耗管理；
- 10) 支持对芯片内部存储器的自测试与冗余修复，支持边界扫描测试。

1.3 主要技术指标

申威 1621 处理器的主要技术指标如下：

- 1) 64 位字长；
- 2) 支持 8 位、16 位、32 位和 64 位整数运算；
- 3) 支持单精度和双精度浮点运算；
- 4) 支持 256 位的整数和浮点短向量加速计算；
- 5) 支持 64 位虚地址空间（实际有效虚地址宽度为 53 位），支持 48 位物理地址空间。每个核心包含 32KB 的一级指令 Cache、32KB 的一级数据 Cache 和 512KB 的二级 Cache，包含 32 条目的全联想 ITB（Instruction-stream Translation Buffer，指令流地址转换缓冲）和 64 条目的全联想一级 DTB（Data-stream Translation Buffer，数据流地址转换缓冲），支持 8KB 的基本页面，也支持多个基本页面组成的大页；还包含由 512 个条目、4 路组相联结构的基本页面转换缓冲和 32 条目、全相联结构的大页转换缓冲组成的二级 DTB。
- 6) 芯片集成 16 个对称的核心（Core3A），16 个核心分布共享容量为 32MB 的三级 Cache；
- 7) DDR3 存储器接口传输率为 1066~1866Mbps，最大存储器总带宽为 119.424GB/s。芯片支持的主存容量为 8、16、32、64、128 或 256GB；
- 8) 2 个 PCI-E 3.0 标准接口各支持 8 链路，每条链路的传输速率为 8Gbps，总双向有效带宽为



32GB/s; 维护接口的最大传输速率为 25Mbps。

2 核心结构

申威 1621 处理器核心（Core3A）是一个完整的 64 位字长的通用处理器，是构成申威 1621 多核处理器的基础，是运行操作系统和用户程序、实现申威 1621 性能的主体，全芯片包含 16 个完全相同的核心。

申威 1621 处理器核心由指令部件、整数执行部件、浮点执行部件、数据 Cache 控制部件、二级 Cache 控制部件、指令 Cache、数据 Cache 和二级 Cache 等部分组成。

2.1 指令部件

指令部件的主要功能是控制程序运行，并处理各种中断和异常。指令部件由转移预测器（BPU）、指令流地址转换（ITB）、取指控制（ICU）、指令译码（FBU）、寄存器重命名（MAP）、指令发射（ISSUE）、重排序缓冲（ROB）和中断和异常管理（IMU）组成。指令部件的结构如图 2-1 所示：

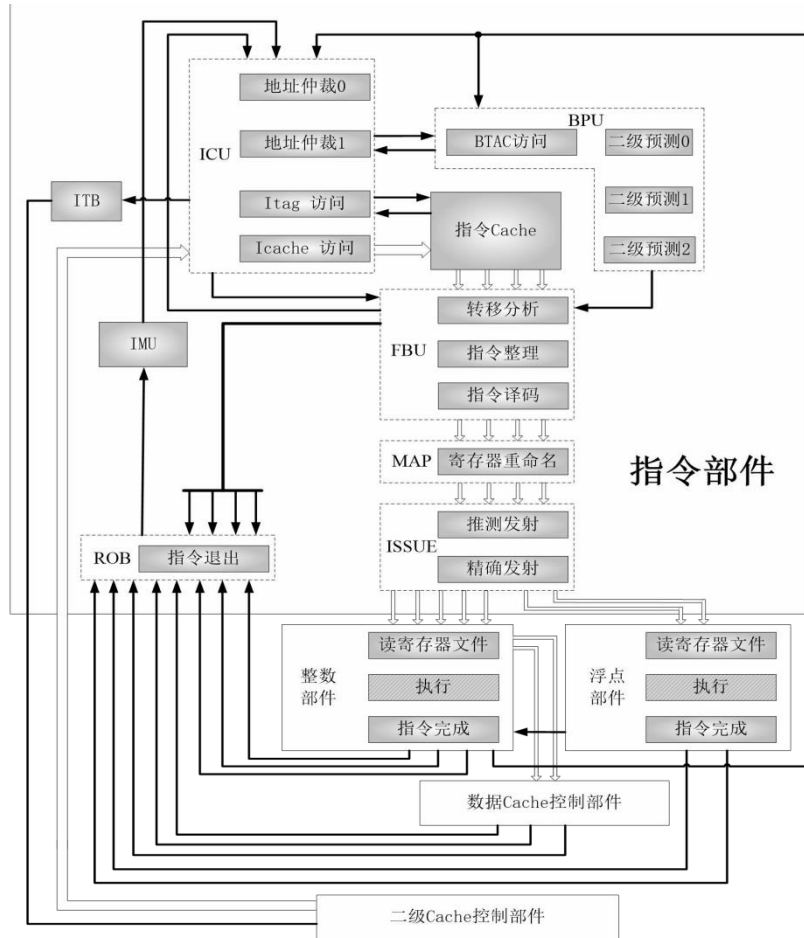


图 2- 1: 申威 1621 处理器指令部件结构图

2.1.1 转移预测器

转移预测器的功能是在取指令时，根据转移指令以及跳转指令的历史信息，预测转移指令或跳转指令的方向和下一条指令地址。Core3A 核心的转移预测器采用两级结构。第一级采用容量较小的转移目标地址缓冲（BTAC），用于快速产生条件转移和无条件转移指令的目标地址预测。第二级则针对不同的转移指令类型分别设置了不同的预测器：对于条件转移指令的方向预测，使用 gshare 和 bimodal 的混合预测器；对于间接跳转指令，设置间接转移目标缓冲（iBTB）进行地址预测；对于返回指令，使用返回地址堆栈（RAS）进行地址预测。

2.1.2 指令流地址转换

指令流地址转换是通过查询 ITB 来实现的，根据指令流当前所处的处理器模式、地址类型以及当前的虚拟机号和进程号，将指令流的虚地址转换成物理地址，并通过权限检查实现存储空间保护功能。当地址转换过程中发生 ITB 脱靶或越权访问时，产生异常自陷来通知操作系统进行处理。

ITB 采用全相联结构，共 32 个条目，并采用轮转替换策略。同时提供多种 ITB 刷新支持，操作系统可以刷新 ITB 指定条目，也可刷新部分或全部 ITB 条目。

2.1.3 取指控制

取指控制包含指令流水线的初始三级站台，其主要功能：

- 1) 控制指令流水线的启动和停止；
- 2) 控制转移预测器的装填和读访问，接收转移预测部件返回的预测信息，并根据该信息进行指令地址预测；
- 3) 选择下一条指令地址，并启动从指令 Cache 中取指令操作；
- 4) 在取指令操作不命中指令 Cache 时，请求 ITB 进行指令流虚实地址转换并向二级 Cache 发出取指令请求；
- 5) 协同中断和异常管理部件管理指令流水线上的中断和异常。

2.1.4 指令译码

指令译码包含三级指令流水线站台。其主要功能：

- 1) 检查指令操作码类型和访问权限的合法性，并根据操作码和预译码信息生成指令执行的功能部件号，对同时能在整数执行部件中执行的指令分配功能部件号，并形成指令执行所需的控制信息和数据信息；
- 2) 根据指令类型形成源寄存器和目标寄存器信息；
- 3) 对系统寄存器进行逻辑寄存器重命名。

2.1.5 寄存器重命名

寄存器重命名包含一级指令流水线站台。其主要功能是对整数寄存器和浮点寄存器分别进行重命名，为源寄存器查询最新映射的物理寄存器，为目标寄存器分配空闲的物理寄存器，消除寄存器间的“读后写”和“写后写”相关性，保证寄存器间的“写后读”真相关性。回收分配出去但已无用的物理寄存器。

2.1.6 指令发射

指令发射包含两级指令流水线站台。在指令发射部件中设置 7 个发射端口，将发射队列分为两级，第一级为等待队列，按整数和浮点指令分类设置；第二级为发射队列，按发射端口分类设置。指令发射部件先从等待队列中筛选出源寄存器可能准备好的指令进入发射队列，然后再从发射队列中挑选满足发射条件的指令送执行部件。

指令可发射条件包括源寄存器是否就绪、执行部件是否存在冲突、写目标寄存器文件端口是否存在冲突、访存指令是否存在冲突等。

指令发射宽度为 7 条指令，每个时钟周期最多可向整数执行部件发射 5 条指令（其中 3 条为整数指令，2 条为访存指令），向浮点执行部件发射 2 条指令。当指令可发射时，将指令的相关信息分别从指令发射队列中选出并送至不同的执行部件。

2.1.7 重排序缓冲

重排序缓冲（ROB: Re-Order Buffer）控制指令的执行顺序，其主要功能是按指令在程序中的顺序，缓存从寄存器重命名站台到执行站台之间正在执行的指令及其相关信息，接收指令在执行期间遇到的故障类型与执行完成信息等，并按程序中的顺序执行指令的退出和异常检查操作。

每个时钟周期重排序缓冲最多可正常退出 4 条指令。从 ROB 的队头开始依次检查，如果指令正常完成，则执行指令退出操作，释放指令占用的 ROB 条目，通知寄存器重命名站台释放相应的物理

寄存器。

如果 ROB 队头的第一条指令发生异常，则向指令流水线和访存流水线广播清空流水线的操作请求，并将异常指令保留的 RAS 栈顶指针送至转移预测器，用于恢复 RAS 的栈顶指针，还要将异常指令的地址和故障信息送至中断和异常管理部件，用于选择异常处理程序的入口地址并写对应的状态与控制寄存器（CSR: Control and status Register）。

2.1.8 中断和异常管理

中断和异常管理部件管理指令部件相关的 CSR 寄存器，执行对 CSR 的显式读、显式写、隐式读和隐式写操作。其主要功能：

- 1) 汇集所有的异常请求及其类型、断点，记录在相应寄存器中，根据异常类型选择相应处理程序的入口；并将该入口地址送至取指控制部件，请求重新取指；
- 2) 汇集所有的中断请求，包括核心内部产生的中断以及系统接口和其它核心产生的中断，经过屏蔽控制和中断使能判断，向取指控制部件提出中断请求。接收取指控制部件保留的断点，并根据中断类型选择相应的中断处理程序入口，将该入口地址送至取指控制部件，请求重新取指令执行；
- 3) 汇集指令部件所有模块的监测状态，根据系统接口中的维护接口产生的状态选择信号，选择对应的状态监测信息输出给外部系统，用于申威 1621 的调试与运行监测。

2.2 整数执行部件

整数执行部件的主要功能是接收发射站台产生的指令执行请求，从整数寄存器文件中读取源操作数，执行相应整数运算操作，将结果写入整数寄存器文件，并向重排序缓冲报告执行完成情况。整数执行部件由整数寄存器文件、3 条整数执行流水线（PipeAL、PipeAU、PipeBU）和两条访存流水线（PipeAM 和 PipeBM）以及访存数据处理模块（LD/STD）组成，其结构如图 2-2 所示。

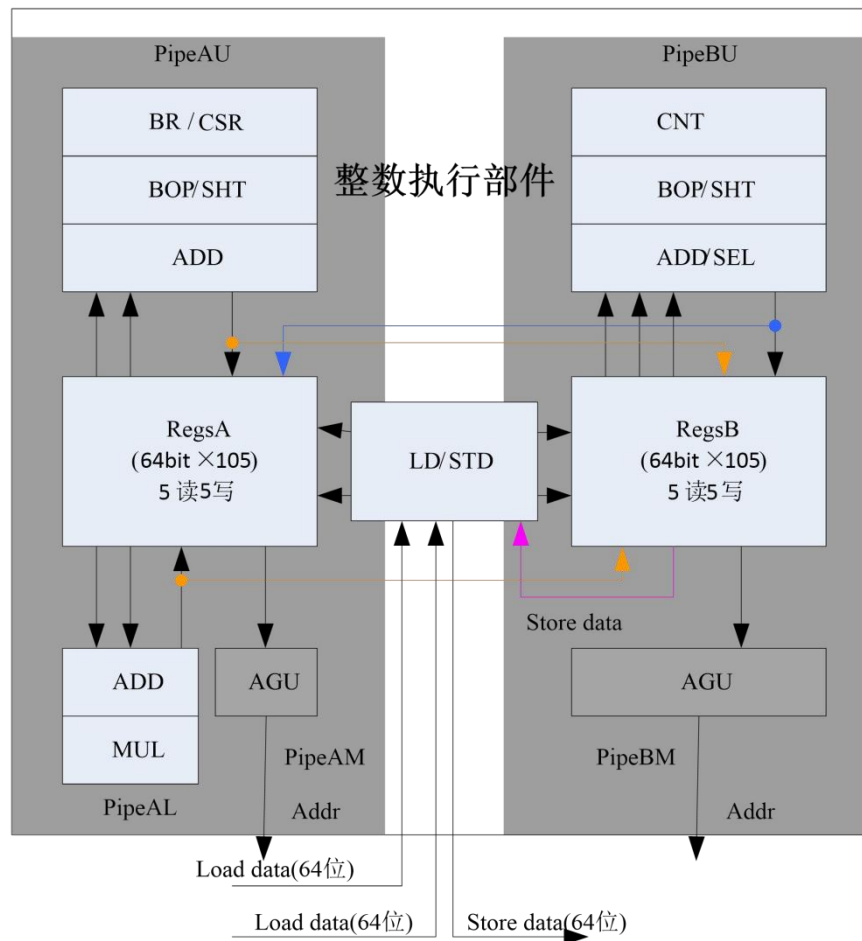


图 2- 2： 整数执行部件结构图

2.2.1 整数寄存器文件

整数寄存器文件由两套各 105 个 64 位寄存器组成（RegsA 和 RegsB），用于保留整数运算的中间结果。其中 31 个对应到指令系统中的整数寄存器，10 个映射到硬件模式下的系统寄存器，还有 64 个作为空闲的物理寄存器用于寄存器重命名。

整数寄存器文件 RegsA 被整数执行流水线 PipeAL、PipeAU 和访存流水线 PipeAM，以及数据处理模块 LD/STD 共同使用，整数寄存器文件 RegsB 被整数执行流水线 PipeBU 和访存流水线 PipeBM，以及访存数据处理模块 LD/STD 共同使用。整数寄存器文件包含 5 个读端口和 5 个写端口，采用旁路技术，实现对同一寄存器的同时写入和读出功能。

2.2.2 整数功能部件

整数功能部件从结构上分为六个部分，即 3 条整数执行流水线（PipeAL、PipeAU、PipeBU）和两条访存流水线（PipeAM 和 PipeBM）以及访存数据处理模块（LD/STD）。其中 PipeAL 包



括加法

子部件（ADD）和乘法子部件（MUL）；PipeAU 包括加法子部件（ADD）、字节操作与移位子部件（BOP/SHT）、转移与控制寄存器访问子部件（BR/CSR）；PipeBU 包括加法与选择子部件

（ADD/SEL）、字节操作与移位子部件（BOP/SHT）、计数子部件（CNT）；PipeAM 和 PipeBM，负责访存指令的地址计算；访存数据处理模块，负责从数据 Cache 向寄存器文件装入数据，或者将寄存器文件中数据写入数据 Cache。整数执行部件每个时钟周期可处理两条整数运算类指令。

2.3 浮点执行部件

浮点执行部件的主要功能是接收发射站台产生的浮点指令执行请求，从浮点寄存器文件中读取源操作数，执行相应浮点和 SIMD 运算操作，将结果写入浮点寄存器文件，并向重排序缓冲报告执行完成情况。浮点执行部件由浮点寄存器文件、浮点主流线、浮点从流水线、浮点访存子部件和 SIMD 整理子部件等组成。浮点执行部件内部共设置了 2 条浮点/SIMD 流水线（U 流水线和 L 流水线），其结构如图 2-3 所示。

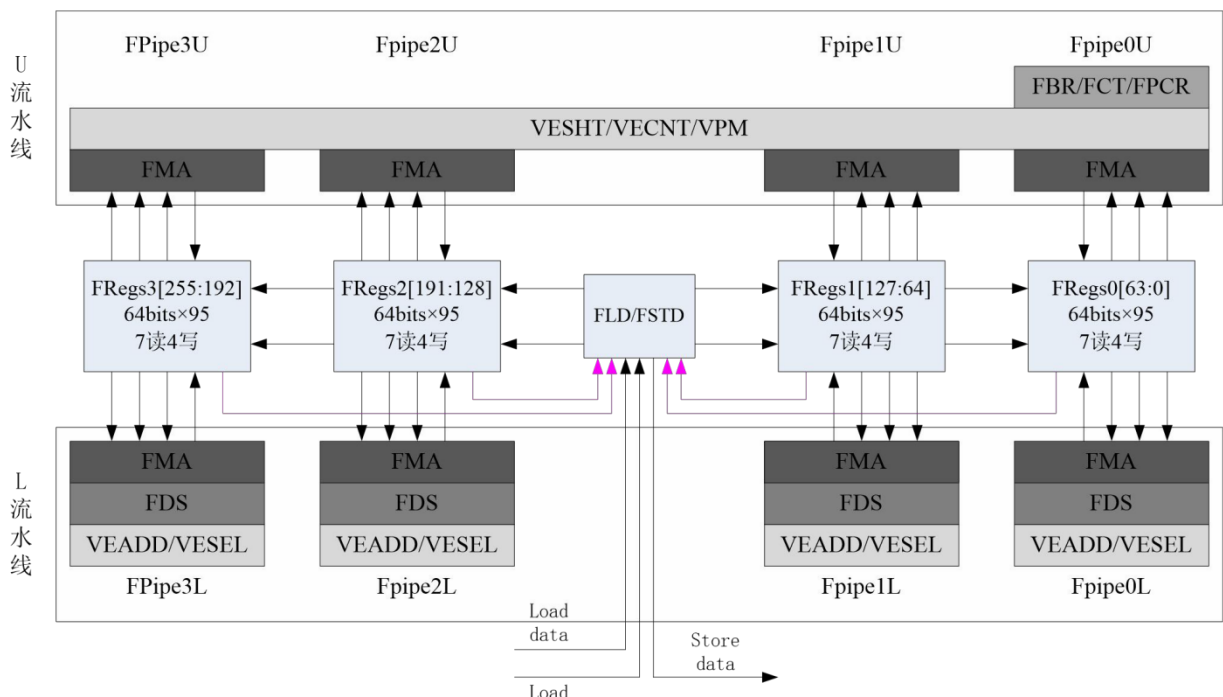


图 2- 3：浮点执行部件结构图

浮点执行部件的特点如下：

- 1) 包含 2 条浮点/SIMD 流水线；
- 2) 支持 IEEE 单精度（简称 S 浮点）和双精度（简称 D 浮点）浮点数据类型，支持 IEEE 所规定的四种舍入模式，并支持各种精确浮点算术异常；
- 3) 支持 SIMD（Single Instruction Multiple Data，单指令流多数据流）短向量运算，短向量类



型 包括字节向量 (8 位×32)、半字向量 (16 位×16)、字向量 (32 位×8)、长字向量 (64 位)

- ×8)、单精度浮点向量（32 位×4）和双精度浮点向量（64 位×4）；
- 4) 支持浮点加、减、乘、除和开平方运算，支持浮点乘加复合运算；
- 5) 支持整数和浮点的短向量运算；
- 6) 支持八倍字整数（256 位）的移位、数“1”、数“0”以及可重构逻辑运算；
- 7) 浮点主流流水线既可以单独处理单精度和双精度浮点运算，又可以与三条从流水线一起共同处理整数和浮点 SIMD 运算。

2.3.1 浮点寄存器文件

2 条浮点/SIMD 流水线共享浮点寄存器文件，该寄存器文件由主从结构的四组寄存器文件组成，即一组主寄存器文件和三组从寄存器文件。每组寄存器文件包含 95 个 64 位寄存器，其中 31 个对应到指令系统中的浮点寄存器，其它 64 个作为空闲的物理寄存器用于寄存器重命名。浮点寄存器文件包含 7 个读端口和 4 个写端口，采用旁路技术，实现对同一寄存器的同时写入和读出功能，以提高浮点执行部件的性能。

2.3.2 浮点功能部件

从结构上看，浮点功能部件包括 2 条浮点/SIMD 流水线和浮点访存子部件。每条浮点流水线分为一条浮点主流流水线和三条浮点从流水线。其中，U 流水线由 FPipe0U、FPipe1U、FPipe2U 和 FPipe3U 组成，L 流水线由 FPipe0L、FPipe1L、FPipe2L 和 FPipe3L 组成。两条流水线上都设有浮点乘加子部件（FMA），仅 U 流水线设有整数向量移位子部件（VESHT）、整数向量计数子部件（VECNT）和整数整理子部件（VPM，处理整数整理 SIMD 和可重构指令），仅 L 流水线设有浮点除法子部件

（FDS，处理浮点除法和平方根指令）、整数向量加法子部件（VEADD）、整数条件选择子部件

（VESEL）。U 流水线和 L 流水线上的 Pipe0 都支持 64 位浮点标量运算。此外，U 流水线的 Pipe0 部分还设有转换子部件（FCT）、转移子部件（FBR）、浮点控制寄存器子部件（FPCR），这些子部件仅支持 64 位浮点标量运算。

2.4 数据 Cache 控制部件

数据 Cache 控制部件管理核心的数据 Cache，其功能是接收来自执行部件的访存请求以及来自二级 Cache 的一致性查询请求和数据装填请求，控制访存请求、数据装填请求和一致性查询请求对数据 Cache 的读写访问，控制访存指令的乱序执行。数据 Cache 控制部件由装入队列



(LQ)、存储

申威 1621 处理器结构手册

队列（SQ）、不命中地址文件（MAF）、数据流地址转换缓冲（DTB）、数据 Cache 访问流水线和 查询处理部件等组成，其结构如图 2-4 所示。

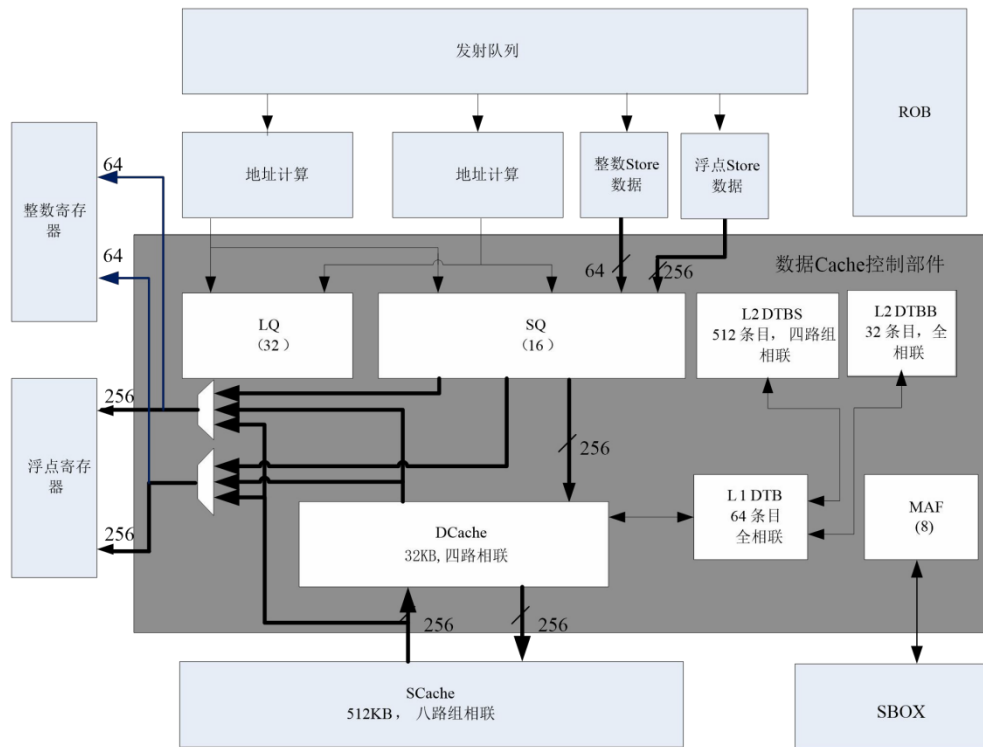


图 2-4: 数据 Cache 控制部件结构图

数据 Cache 控制部件的特点如下：

- 1) 支持两条访存流水线对一级数据 Cache 的并行、无阻塞访问；
- 2) 与整数寄存器文件之间的数据通路为 64 位；
- 3) 与浮点寄存器文件的数据通路为 256 位；
- 4) 通过数据旁路、请求重试、重发自陷等方式处理访存指令间的相关性，支持访存指令的乱序执行和推测执行；
- 5) 通过特殊的一级 Cache 访问流水线设计，解决了一级数据 Cache 结构冲突（包括访问端口冲突和存储体冲突），采用双端口物理存储器作为 2 读 1 写的虚拟三端口 Cache 存储器；
- 6) 支持多 Cache 数据一致性处理；
- 7) 通过两级 DTB，降低数据流虚实地址转换脱靶的概率。

2.4.1 装入队列

装入队列（LQ: Load Queue）是一个用于处理装入指令（Load）的重排序缓冲，包含 32 个条目，保存已发向数据 Cache 控制部件的 Load 指令的相关状态，直到 Load 指令被中止或者需要的数据



返

回且该指令可以退出（Retire），才从 LQ 中撤销。

LQ 控制 Load 指令的乱序执行，检查访存指令间的相关性冲突，保证一级数据 Cache 无阻塞访问的正确性。

2.4.2 存储队列

存储队列（SQ: Store Queue）是一个用于处理存储指令（Store）的重排序缓冲，包含 16 个条目，保存已发向数据 Cache 控制部件的 Store 指令的相关状态，直到 Store 指令被中止或者写数据写入一级数据 Cache，才从 SQ 中撤销。在 Store 指令退出（Retire）前，SQ 保存 Store 指令的相关数据，Store 指令退出（Retire）后才允许将数据写入一级数据 Cache。

SQ 控制 Store 指令的乱序执行，检查访存指令间的相关性冲突，保证一级数据 Cache 无阻塞访问的正确性。SQ 支持 Load 指令旁路相同地址且年老的 Store 指令的数据，可缩短平均访存延时。

2.4.3 不命中地址文件

不命中地址文件（MAF: Miss Address File）是一个访存请求的悬挂缓冲，包含 8 个条目，用来保存下列请求的物理地址及相关信息：

- 1) Load 或 Store 指令不命中数据 Cache 而产生的一级数据 Cache 装填请求；
- 2) Store 指令命中一级数据 Cache，但 Cache 行状态为非可写状态而产生的请求；
- 3) I/O 空间的 Load 指令产生的 I/O 空间读请求。

2.4.4 数据流地址转换缓冲

数据流地址转换是通过查询数据流地址转换缓冲（DTB: Dstream Translation Buffer）来实现的，根据指令流当前所处的模式、地址类型及当前的虚拟机号和进程号，把数据流的虚地址转换成物理地址，并通过权限检查实现存储空间保护功能。DTB 采用两级结构（L1DTB 和 L2DTB），当 L1DTB 脱靶时，继续查询 L2DTB，如果命中，则自动装填 L1DTB，否则将产生 DTB 脱靶。当地址转换过程中发生 DTB 脱靶或越权访问时，产生异常自陷来通知操作系统进行处理。

一级 DTB（L1DTB）共 64 个全相联条目，并采用轮转替换策略；二级 DTB（L2DTB）包含 512 个条目、4 路组相联结构的基本页面转换缓冲和 32 个条目、全相联结构的大页转换缓冲。同时提供多种 DTB 刷新支持，操作系统可以刷新 DTB 指定条目，也可刷新部分或全部 DTB 条目。

2.4.5 一级数据 Cache 访问流水线

一级数据 Cache 访问流水线分为流水线 0、1、2 三条，负责控制各种请求对双端口一级数据 Cache

存储器的访问，实现虚拟三端口一级数据 Cache 的功能。

流水线 0 和流水线 1 负责处理整数和浮点执行部件产生的访存请求和 LQ 及 SQ 产生的重试请求。这些请求查询一级数据 Cache 标记存储器，确定是否命中一级数据 Cache，同时还需要查询 LQ、SQ 和 MAF 这 3 个缓冲，通过与上述三个缓冲中保存的访存指令进行年龄和地址比较，确定是否存在相关性冲突，并通过冲突重试和重发自陷等方式处理数据相关性冲突。不存在冲突的请求，将根据对应指令的类型和命中一级数据 Cache 的情况，进行后续处理。访存请求和重试请求通过流水线 0、1 获得对一级数据 Cache 端口的使用权，同时还获得整数和浮点寄存器文件用于写入“命中”一级数据 Cache 而返回数据的写端口使用权。

流水线 2 负责处理一级数据 Cache 装填请求和 SQ “读-修改-写”请求。装填请求通过流水线 2 获得一级数据 Cache 端口的使用权，同时还获得整数和浮点寄存器文件用于写入“不命中”一级数据 Cache 而返回数据的写端口使用权。

2.4.6 查询处理部件

查询处理部件是针对二级 Cache 控制部件发来的一致性查询请求而设置的，辅助实现一级数据 Cache 与二级 Cache 之间，以及不同核心二级 Cache 之间的数据一致性。该部件采用非流水方式处理查询请求。对于需要获得最新数据的一致性查询请求，该部件使用数据 Cache 端口 1 将数据读出，并传送给二级 Cache 控制部件。

2.5 二级 Cache 控制部件

二级 Cache 控制部件实现核心对外接口的控制和二级 Cache 的控制，处理核心指令流和数据流对二级 Cache 的访问，并控制核心与核组的一致性处理部件之间各类请求的传输。二级 Cache 控制部件由不命中地址文件（SMAF）、淘汰地址文件（SVAF）、淘汰数据缓冲（SVDB）和二级 Cache 访问流水线等组成。

二级 Cache 控制部件的特点如下：

- 1) 支持对二级 Cache 的无阻塞访问；
- 2) 二级 Cache 和一级指令 Cache 以及数据 Cache 之间的数据通路为 128 位；
- 3) 通过请求重试的方式处理数据相关性冲突和结构相关性冲突；

- 4) 通过资源预分配（包括 SMAF、SVAF 和 SVDB）和资源及时释放策略，处理资源竞争问题，实现无“死锁”设计；
- 5) 支持基于 MESI 和 MOESI 协议的多 Cache 一致性处理。

2.5.1 不命中地址文件

二级 Cache 控制部件中也包含一个不命中地址文件（SMAF），共有 8 个条目，用来保存发向核心外部请求的物理地址和相关信息，具体有：

- 1) 指令流取指令或指令预取产生的、不命中二级 Cache 的指令读请求；
- 2) 数据流 Load/Store 类指令产生的、不命中二级 Cache 的数据读请求；
- 3) 数据流 Store 类指令产生的、命中二级 Cache 且为非可写状态的置脏请求。

2.5.2 淘汰地址文件和淘汰数据缓冲

淘汰地址文件(SVAF)和淘汰数据缓冲(SVDB)包含作用不同的两套（SVAF1 与 SVDB1、SVAF2 与 SVDB2），每套分别包含 8 个条目，其中 SVAF1 和 SVAF2 存放地址和控制相关的信息，SVDB1 和 SVDB2 存放数据，其每个条目都可以存放 128 字节的数据。SVAF 和 SVDB 的作用如下：

- 1) 数据 Cache 中被淘汰的、状态为“已修改”的 Cache 行，需要回写到二级 Cache，则回写请求和对应的 Cache 行数据块，保存在 SVAF1 和 SVDB1 中；不能回写到二级 Cache，而需要回写到外部存储器时，则保存在 SVAF2 和 SVDB2 中；
- 2) 来自外部存储器，需要装填到二级 Cache 的装填请求和对应的 Cache 行数据块，保存在 SVAF1 和 SVDB1 中；
- 3) 二级 Cache 被淘汰的、状态为“脏”的 Cache 行，需要回写到主存，则回写请求和对应的 Cache 行数据块，保存在 SVAF2 和 SVDB2 中；
- 4) 处理来自核心外部实现 Cache 一致性的查询请求，在处理过程中将被悬挂在 SVAF2 中，如果查询请求命中核心内 Cache 且需要获得 Cache 中数据，则数据保存在 SVDB2 中。

2.5.3 二级Cache 访问流水线

二级 Cache 访问流水线负责处理本核心产生的流请求（包括数据流请求和指令流请求）以及核心外部一致性处理部件产生的一致性查询请求和响应。

进入流水线的请求，将获得对二级 Cache 的访问权限，并通过查询二级 Cache 标记存储器，
确

定是否命中二级 Cache，同时还需要查询 SVAF1、SVAF2 和 SMAF 这三个缓冲，通过与上述三个缓冲中保存的请求信息进行比较，确定是否存在相关性冲突，并通过阻塞请求或重试等方式处理冲突。

二级 Cache 控制部件对各类请求进行流水处理，一般每 4 个时钟周期能接收并处理一个请求。为缩短存储器访问延时，优先处理响应请求。各类请求必须在预分配到所有可能使用的缓冲资源的情况下，才被允许进入流水线。在流水线处理过程中，一旦发现某些缓冲资源无需使用，则立即释放，保证后继请求可以使用这些资源，以避免资源浪费，同时保证不会由于缓冲资源争用而产生请求处理的“死锁”。

3 核心流水线组织

3.1 指令流水线

申威 1621 处理器核心的指令流水线由 13 级站台组成，具体如图 3-1 所示。

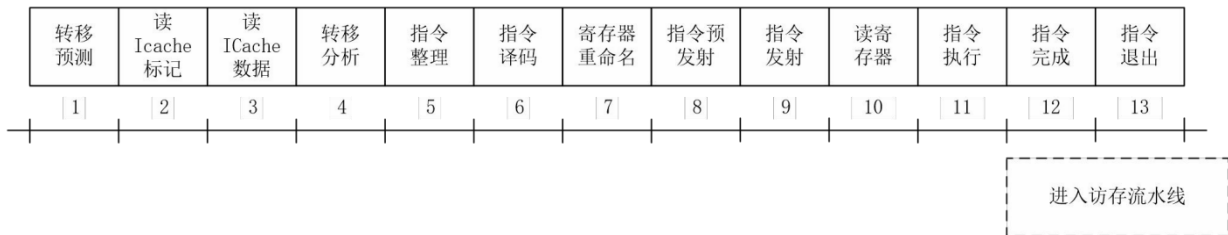


图 3-1：申威 1621 指令流水线结构图

3.1.1 转移预测站台

本站台汇集各路取指令地址以及转移预测结果，根据优先级进行仲裁，并根据当前流水线的状态，启动或停止对指令 Cache 和转移预测部件的访问。取指令地址来源包括程序计数器（PC：Program Counter）、转移预测地址、执行部件对转移预测进行修正的地址以及异常与中断处理程序的入口地址等。

3.1.2 读指令Cache 标记站台

本站台访问指令 Cache 的标记存储器，获取取指地址是否命中指令 Cache 的相关信息。如果命中指令 Cache，则从指令 Cache 取出四条指令送指令流水线；如果不命中，则提交 ITB 进行虚实地址转换，产生指令流请求送二级 Cache 控制部件，并启动指令预取。

3.1.3 读指令Cache 数据站台

本站台访问指令 Cache 的数据，获取需要执行的指令。从指令 Cache 取出四条指令，与 ICache 装填缓冲中指令进行选择并送取指缓冲。

3.1.4 转移分析站台

本站台从取指缓冲中依次读出指令进行转移分析（一次最多分析四条指令），根据指令类型进行地址计算或查询返回地址堆栈，根据需要向取指部件发送改道请求和地址；根据转移发生情况，

控制后续指令有效位；并识别空指令并清除这些指令的有效位。

3.1.5 指令整理站台

本站台负责识别特殊指令，选择可以同时送后续站台的指令（一次最多送四条指令），并对指令的源和目标寄存器进行译码，识别非法操作码和各种指令屏蔽异常。

该站台为每条指令分配唯一的指令序列号（Inum），该序列号指定了指令在重排序缓冲（ROB）中的位置，也用于标识指令在程序中的先后次序（即指令年龄的大小）。

3.1.6 指令译码站台

本站台负责对指令整理站台选出的指令（一次最多四条指令）进行译码，静态分配执行子部件号，分析寄存器相关性并形成标志。另外，为了保证隐式读写和显式读写核心内部控制与状态寄存器（CSR）指令之间的顺序，还判断和设置了 CSR 的记分板，以控制指令进入后续流水线的时刻。

3.1.7 寄存器重命名站台

本站台负责对整数和浮点指令分别进行整数寄存器重命名和浮点寄存器重命名。通过给指令中的目标寄存器分配一个新的物理寄存器，来消除指令之间的寄存器读后写（WAR: Write After Read）或写后写（WAW: Write After Write）相关性；通过为指令的源寄存器指定其对应的物理寄存器号，保持指令之间的寄存器写后读（RAW: Read After Write）相关性。

本站台根据指令类型和当前流水线的状态，按指令在程序中的顺序，分别将整数和浮点指令压入整数预发射队列和浮点预发射队列。

3.1.8 指令预发射站台

本站台负责预判断指令的发射条件，并从源寄存器可能满足发射条件的指令中挑选并送至指令发射队列。

3.1.9 指令发射站台

本站台负责精确判断指令的发射条件，并从所有满足发射条件的指令中挑选年龄最老的发向后续站台。每个时钟周期，最多可从整数发射队列中挑选 3 条整数指令发向整数执行部件的 3 条整数

执行流水线（PipeAL、PipeAU、PipeBU），从访存发射队列中挑选 2 条访存指令发向整数执行部件的 2 条访存流水线（PipeAM 和 PipeBM），从浮点发射队列中挑选 2 条浮点指令发向浮点执行部件 2 条浮点/SIMD 流水线（U 流水线和 L 流水线）。

指令发射逻辑采用记分板机制保证寄存器的真相关性（写后读相关），同时还需要检查功能部件是否冲突、写寄存器文件是否冲突、访存队列是否满等条件。

3.1.10 读寄存器站台

本站台用于形成执行部件所需的源操作数。源操作数的来源包括从寄存器文件中读出的数值、执行部件中各功能子部件的运算结果、数据 Cache 控制部件返回的访存数据以及指令中携带的立即数或偏移量，具体需要根据指令的类型和数据返回的时刻进行选择。另外，该站台还需要为执行部件提供相关的控制信息。

各功能部件的运算结果一般写入目标寄存器，下一个时钟周期才能读出使用。整数执行部件和浮点执行部件都支持结果数据旁路功能，即读寄存器文件时，如果所读的寄存器正在被写入，则可直接使用写入的数据，缩短运算结果从产生到再次使用的延时。

3.1.11 执行站台

本站台的执行延迟不固定，不同的指令类型有不同的操作延迟。除浮点除法和浮点平方根指令外，其它指令执行都支持流水操作，即每个时钟周期可接收一条新指令，也可以产生一个新的运算结果。

对于转移类指令，本站台要检查并确定转移预测是否成功。整数执行部件和浮点执行部件的转移子部件根据指令类型，判断是否进行转移以及预测的转移目标指令地址是否正确。如果发现转移预测失败（转移方向预测失败或者转移目标预测失败），要立即清空流水线，废除转移指令后取出的所有指令，重新取指令执行。

对其它非访存类指令，在执行站台处理的最后一个周期，根据需要将运算结果写入目标寄存器中。

对访存指令，该站台主要计算指令的访存虚地址，然后就进入访存流水线，通过 DTB 进行虚实地址转换，产生数据流的物理地址，然后查询数据 Cache。当装入指令所需的数据返回时，通知发射逻辑解除对应记分板，将获得的数据写入目标寄存器。

3.1.12 指令完成站台

本站台负责将非访存类指令的执行结果写入重排序缓冲（ROB），等待指令退出。对于访存类指令，当装入操作和存储操作结束时，将执行结果状态写入重排序缓冲，等待指令退出。

3.1.13 指令退出

本站台每个周期从重排序缓冲头部读出 4 条指令信息，如果指令已经执行完成且没有发生异常，则允许指令退出，通知寄存器映射站台释放占用的物理寄存器，将指令从重排序缓冲中删除，指令处理完成。如果指令发生异常，则清空流水线，通知各流水线站台进行修复和清除操作，并将指令的异常信息送异常和中断管理部件进行处理。

3.2 运算流水线

在整数和浮点执行部件内处理的指令，需要经过 3 级流水站台，分别是读寄存器、执行和结果写回。其中读寄存器和结果写回的操作延时都是 1 个时钟周期，而执行站台则根据指令类型的不同有不同的操作延时。

申威 1621 包含 3 条整数运算流水线（PipeAL、PipeAU、PipeBU），分别对应整数执行部件中（E0、E1 和 E2），3 个部件都由多个子部件组成，部件 0、1、2 中有相同的子部件，也有不同的子部件。整数运算流水线都支持流水操作，多数整数运算指令的操作延时为一个时钟周期，整数乘法运算指令的操作延时为 4 个时钟周期，读 CSR 寄存器指令的操作延时为 3 个时钟周期，数“1”指令的操作延时为 2 个时钟周期。

申威 1621 包含 2 条浮点/SIMD 运算流水线，均由一条主流水线和三条从流水线组成，主流水线既可以单独处理单精度和双精度浮点运算，也可以与从流水线配合处理整数和浮点 SIMD 运算。浮点运算流水线除浮点除法和浮点平方根外，都支持流水操作。浮点运算指令（包括 SIMD 浮点指令）的操作延时一般为 6 个时钟周期，一些简单指令的操作延时为 1~2 个时钟周期，浮点除法和浮点平方根不支持流水操作，单精度浮点的操作延时分别 17 个时钟周期，双精度浮点的操作延时分别为 32 和 31 个时钟周期。整数 SIMD 指令的操作延时一般为 1~2 个时钟周期，只有八倍字处理指令（SLOW、SRLW 和 CTPOPOW）的操作延时为 3 个时钟周期。

3.3 访存流水线

对 Load/Store 类访存指令，在指令流水线的读寄存器站台之后进入访存流水线，该流水线的长度与 Load/Store 类指令是否命中 Cache 相关。当命中一级数据 Cache 时，访存流水线有 4 级站台，具体如下：

- 1) 地址计算和 DTB 转换：计算出 Load/Store 类指令的有效虚地址，并通过 DTB 转换为物理地址；
- 2) 查询数据 Cache：用转换后的物理地址查询数据 Cache，判断是否命中数据 Cache；
- 3) 读数据 Cache 和数据传输：对 Load 类指令，读出数据 Cache 中“命中”的数据，并通过内部数据通路传输给整数或浮点执行部件。对 Store 类指令，将写数据通过内部数据通路传输给存储队列 SQ；
- 4) 写寄存器文件：对 Load 类指令，将内部数据通路上的数据，经过转换后写入整数寄存器文件或浮点寄存器文件。对 Store 指令，将内部数据通路上的数据放入存储队列 SQ。

对 Load/Store 类指令，由于数据流访问可能命中核心内的一级数据 Cache，也可能命中核心内的二级 Cache，还可能需访问外部存储器，因此操作延迟是不固定的。若命中一级数据 Cache，则从读寄存器到结果写入目标寄存器的延时为 4 个时钟周期。若命中二级 Cache，则延时最短为 11 个时钟周期。

对整数寄存器文件与浮点寄存器文件之间的数据传输指令，其操作延时为 4 个时钟周期。

4 芯片结构

申威 1621 处理器是一种集成处理器核心、存储器和 I/O 的片上系统 (SoC) 处理器, 既包含 16 个核心, 还集成 4 个一致性处理部件、32MB 分布共享三级 Cache、8 通道存储控制器和对应的 DDR3 存储器接口, 同时集成了包含 2 个 PCI-E 接口、维护接口在内的系统接口。上述模块通过“片上环网”实现异步互连。存储控制器通过申威 1621 的 DDR3 存储器接口连接外部的 DDR3 存储器。

4.1 片上互连结构

片上集成的 16 个核心 (分为 4 个核组)、4 个一致性处理部件 (各包含 8MB 的三级 Cache 分体)、8 路 DDR3 存储器接口、2 个 PCI-E 接口和集中控制部件通过 4 套“双向环网连接”, 如图 1-2 所示。环网包括 6 个网络节点, 最长跳步数 (hop) 为 3。核组#0~#3 连接环网节点 #0~#3, PCI-E#0~#1 连接环网节点#4~#5。为实现网络的对称设计, 集中控制部件可以通过配置方式与环网节点#4 或#5 连接, 核心必须通过该环网节点访问集中控制部件 (IO 请求、中断等)。环网节点#4 或#5 由 IO 路由单元 (IRU, IO Routing Unit) 和 IO 节点单元 (ION, IO Node) 组成。

4.2 共享存储结构

核组中的 16 个核心通过片上环网实现互连, 共享三级 Cache 和由存储控制器控制的主存, 构成一个完整的 CC_NUMA 结构的多处理器子系统, 通过 4 个一致性处理部件实现 16 个核心的多 Cache 一致性。

在结构设计上, 16 个核心在操作系统控制下可进行动态裁剪, 被裁剪的核心可使其处于“睡眠”状态, 以降低其运行功耗。通过增加“活动”的核心数量可以有效提高处理性能, 减少“活动”的核心数量可以相对提升“活动”核心的主存带宽和主存容量。

核组内采用多时钟设计, 其中核心时钟频率最高, 一致性处理部件的时钟频率次之, 存储控制器采用了较低的时钟频率, 以降低运行功耗。不同时钟之间通过数据通路宽度的调整, 实现数据传输带宽的匹配。

4.3 一致性处理部件

一致性处理部件负责处理 16 个核心，或系统接口对主存的访问请求，实现基于“三级 Cache 标记存储器”的共享一致性存储器访问。一致性处理部件主要由三级 Cache 分体、一致性处理控制和

存储器访问缓冲（MAQ）组成。

4.3.1 三级Cache

三级 Cache（TCache）容量为 32MB，由 4 个 8MB 的分体组成，每个分体包含标记存储器（TTag）和数据存储器（TData）两部分，其中标记存储器记录本三级 Cache 分体和各核心内部 Cache 的数据块副本的状态，通过状态的查询与处理，可实现芯片内的多 Cache 一致性。

标记存储器中存放的条目信息如图 4-1 所示。其中状态由“V”和“W”两位表示，其中“V”位表示对应核心的 Cache 中保存有效的副本，“W”位表示对应核心的 Cache 中保留的副本是唯一 的最新副本，且本核心拥有该副本的写权限，即“可写”。两位状态位的定义如表 4-1 所示，其中 x 表示可为 0，也可为 1。

TTag	状态(2位)	核心位向量(4位)	TCache-Tag
-------------	---------------	------------------	-------------------

图 4-1: TTag 条目

表 4-1: 标记存储器状态编码

TTag 状态	状态名称	含义
2' b00	无效	TCache 和所有核心都没有该数据的副本。
2' b01	清洁	TCache 拥有该数据的清洁副本，且与主存数据一致。核心位向量记录哪些核心拥有该清洁副本。
2' b10	脏在核心	某个核心拥有该数据的最新副本，与 TCache 和主存数据不一致。
2' b11	脏在 TCache	TCache 拥有该数据的最新副本，且与主存数据不一致。核心位向量记录哪些核心拥有该副本，且与 TCache 一致。

4.3.2 一致性处理控制

一致性处理控制通过“标记存储器”，实现对各种存储器访问请求的处理，处理方式是根据请求的地址和类型，查询三级 Cache 标记存储器，获得请求数据在各核心 Cache 中的分布与“被修改”状况，按照 Cache “无效”一致性处理原则和状态转换关系，修改标记存储器状态，并产生到核心的一致性查询请求（二次请求）、到核心的响应（核组产生的访存请求）以及到系统接口的响应（非核组产生的访存请求）；对最新数据副本在三级 Cache 的请求，则读取三级 Cache 的数据，通过响应返回核心，并修改三级 Cache 标记存储器中的副本状态；对需要访问主存的请求，则将请求送入存储器访问缓冲。另外，对不需要进行 Cache 一致性处理或者访问系统接口的请求，则直接发送到

存储控制器。

4.3.3 存储器访问缓冲

存储器访问缓冲（MAQ: Memory Access Queue）用于存放一致性处理部件对存储器的读写访问请求，一致性处理控制将完成一致性处理的访存请求放入 MAQ，等待完成后继处理（包括访问外部存储器），请求处理完毕则从 MAQ 中删除，并根据 MAQ 中保留的信息产生响应返回请求源。同时 MAQ 还支持对存储器进行“读-修改-写”操作。

4.4 存储控制器

存储控制器的功能是处理来自一致性处理部件的主存访问请求，实现对外部 DDR3 存储器的初始化设置、刷新控制、读写访问控制和功耗管理。存储控制器主要由控制模块、数据通路和存储器接口组成。其特点如下：

- 1) 支持连接符合 JEDEC 标准的各种 DDR3 SDRAM 存储器；
- 2) 支持 1066Mbps、1333Mbps、1600Mbps 和 1866Mbps 等 DDR3 数据传输率；
- 3) 支持连接单 Rank 模式、双 Rank 和 4Rank 模式的外部 DDR3 存储器；
- 4) 单、双 Rank 模式下的外部存储器容量分别为 2GB、4GB、8GB 或 16GB，4Rank 模式下支持 4GB、8GB、16GB 或 32GB；
- 5) 存储器接口采用双通道结构，每条通道宽度为 72 位，包含 64 位数据和 8 位 ECC 校验，可实现对数据纠单错和检多错操作；
- 6) 支持 DDR3 标准定义的各种延迟参数，支持 DDR3 接口的参数测试；
- 7) 支持 DDR3 SDRAM 芯片的浅睡眠和深睡眠。

4.4.1 控制模块

控制模块负责处理与存储控制器相关的外部请求和内部请求以及 ODT（ODT: On-Die Termination，片上端接电阻）控制功能。外部请求包括来自一致性处理部件的存储器读写请求、控制寄存器读写请求等。内部请求包括 SDRAM 存储器刷新请求、自刷新请求、ZQ 请求（用于调整和校准端接电阻）等。ODT 控制功能负责根据片外 DDR3 SDRAM 存储器的工作状态，改变对 SDRAM 片上端接电阻的控制，ODT 控制分为同步 ODT 和动态 ODT 两类。

控制模块处理存储器读/写请求时，支持“细粒度自适应 Open-Page”和“Close-Page”两种存储

器页面管理模式（存储器页面是指同一存储器“体”的同一“行”所对应的存储空间）。在“细粒度自适应 Open-Page”模式下，存储控制器会根据请求缓冲中后续访问是否与当前待发送请求属于同一页面，来决定是否发送不带“自动预充”的读/写命令到 DDR3 存储器，以便在读写操作完成后保持该请求所访问页面的“打开”状态。在“Close-Page”模式下，存储控制器发送带“自动预充”的读/写命令到 DDR3 存储器，以便在读写操作完成后自动关闭该请求所访问页面。软件可以通过配置存储控制器内部的 IOR 寄存器，实现“Open-Page”和“Close-Page”之间的模式切换。

4.4.2 数据通路

数据通路的功能是处理存储器读写数据，包括数据发送和数据卸载两个通路。数据发送通路负责缓冲存储器写数据，并根据 DDR3 存储器接口时序，在写命令发出的指定周期后，将数据发送到 DDR3 存储器接口的数据总线上。数据卸载通路处理读响应数据，并负责消除不同 SDRAM 存储器芯片返回数据之间延时偏斜，保证数据的正确接收。

4.4.3 存储器接口

控制器接口负责连接申威 1621 芯片和外部 DDR3 SDRAM 存储器芯片，接口信号电平采用 SSTL15（1.5V），最高工作频率 800MHz，信号传输率可达 1600Mbps。存储器接口包含连接 DDR3 SDRAM 存储器所必须的时钟和时钟控制信号、双向数据和数据控制信号、地址和命令信号以及复位、信号端接控制信号。

4.5 PCI-E 接口部件

PCI-E 接口部件连接接口处理模块和符合 PCI-E 2.0 规范的标准 PCI-E 接口，实现双方之间请求 格式的转换与控制。PCI-E 接口部件的工作频率为 250MHz，其主要功能特点如下：

- 1) 实现符合 PCI-E 2.0 规范的根节点（RC: Root Complex）；
- 2) 连接×8 全双工高速 PCI-E 链路，在链路初始化时可与连接的设备自动协商链路宽度为×1、×2、×4 或×8；
- 3) 支持 2.5Gbps 或 5.0Gbps 的链路传输速率，有效双向数据传输带宽最高可达 8GB/s；
- 4) 支持 PCI-E 2.0 规范所定义的各种低功耗状态；
- 5) 支持 PCI-E 规范所定义的基本和高级错误报告机制；
- 6) 支持 256 种 MSI-X 中断和 4 种 INTx 中断（即 INTA/B/C/D）；
- 7) 支持单个虚通道，8 种流量类型；
- 8) 支持 PCI-E 链路层流量控制；
- 9) 支持端到端的 CRC 校验；
- 10) 支持基于 IOMMU 的 IO 虚拟化功能。

4.6 集中控制部件

集中控制部件包含维护接口和中断控制两个子部件。

4.6.1 维护接口子部件

维护接口部件连接接口处理模块和自定义的维护接口，实现对申威 1621 处理器的复位、初始化、自引导和维护功能。维护接口部件的工作频率为 25MHz，其主要功能特点如下：

- 1) 处理器复位有上电复位、冷复位以及睡眠复位等三种类型，前两种是通过芯片引脚触发的 复位，第三种是由睡眠中断或维护命令触发的复位；
- 2) 处理器初始化是在复位结束后，通过 SPI 自引导接口，读取片外 Flash 存储器中的相关信息，对申威 1621 进行配置、测试和初始化程序加载。包括配置并启动片内各核心的存储器自测 试或冗余修复，通过自引导接口实现对所有核心的指令 Cache 进行初始化程序加载，以及对 DDR3 存储器接口和 PCI-E 接口的 I/O 寄存器进行配置。同时还能启动加载到核心指令 Cache 中的初始化程序运行，进行更为全面的处理器初始化、配置、测试以及系统引导功能；

- 3) 维护功能是通过维护接口与外部维护系统互连，接收外部维护命令，实现读写申威 1621 内

部的 I/O 寄存器和主存内容，通过向核心发送维护中断，实现更为复杂的维护功能，支持程序调试和系统调试；

- 4) 维护接口部件连接的维护接口，还支持处理器内部工作状态监测，并向外部系统报告处理器初始化状态、故障状态以及维护中断处理状态，并可设置处理器的初始化方式。

4.6.2 中断控制子部件

中断控制子部件负责完成如下功能：

- 1) 收集来自 I/O 接口的中断请求（包括 PCI-E 的 MSI-X 中断和 INTx 中断、睡眠中断、维护中断和故障中断等）以及接口处理部件产生的短时钟中断，转发到相应的目标核组，再由核组发送到目标核心；
- 2) 接收来自核心的核间中断、睡眠中断，作为响应转发到相应目标核心；
- 3) 接收来自核心的异步消息，并通过异步消息中断方式通知目标核心；
- 4) 在核心睡眠和唤醒处理流程中配合维护接口部件完成相应控制。

4.7 时钟接口

申威 1621 的时钟接口通过外部时钟输入和集成在内部的三个时钟锁相环（PLL），为内部电路提供三路工作时钟。其中核心使用频率最高的工作时钟，一致性处理部件和互连使用频率次高的工作时钟，存储控制器使用独立的一路工作时钟，每路工作时钟的频率都可以通过引脚分别进行灵活配置，还支持 PLL 旁路模式支持芯片调试。

PCI-E 接口部件的工作时钟来自 PCI-E 接口，2 个 PCI-E 接口有各自独立的时钟输入和时钟锁相环（PLL），产生 500MHz 的工作时钟。

维护接口部件的工作时钟来自独立的时钟输入，一般时钟频率为 25MHz。

4.8 测试接口

申威 1621 采用完全兼容 IEEE1149.1 标准的 JTAG 接口，实现对芯片的可测性支持。测试接口的主要功能如下：

- 1) 边界扫描测试：支持对芯片引脚进行边界扫描测试；
- 2) 存储器自测试：支持片内存储器的自测试（BIST）和冗余修复测试（BISR），支持片内存储器调试功能；

3) 工艺测试: 内设多种工艺测试电路, 对芯片流片工艺的特性进行检测和测试。

5 存储体系结构

5.1 层次化存储体系结构

申威 1621 处理器的层次化存储体系结构由三级存储层次组成，即寄存器（包括整数寄存器和浮点寄存器）、核心内部的 Cache（一级数据 Cache 与指令 Cache 以及二级 Cache）和芯片级的共享三级 Cache，以及存储控制器控制下的共享主存。具体如图 6-1 所示。

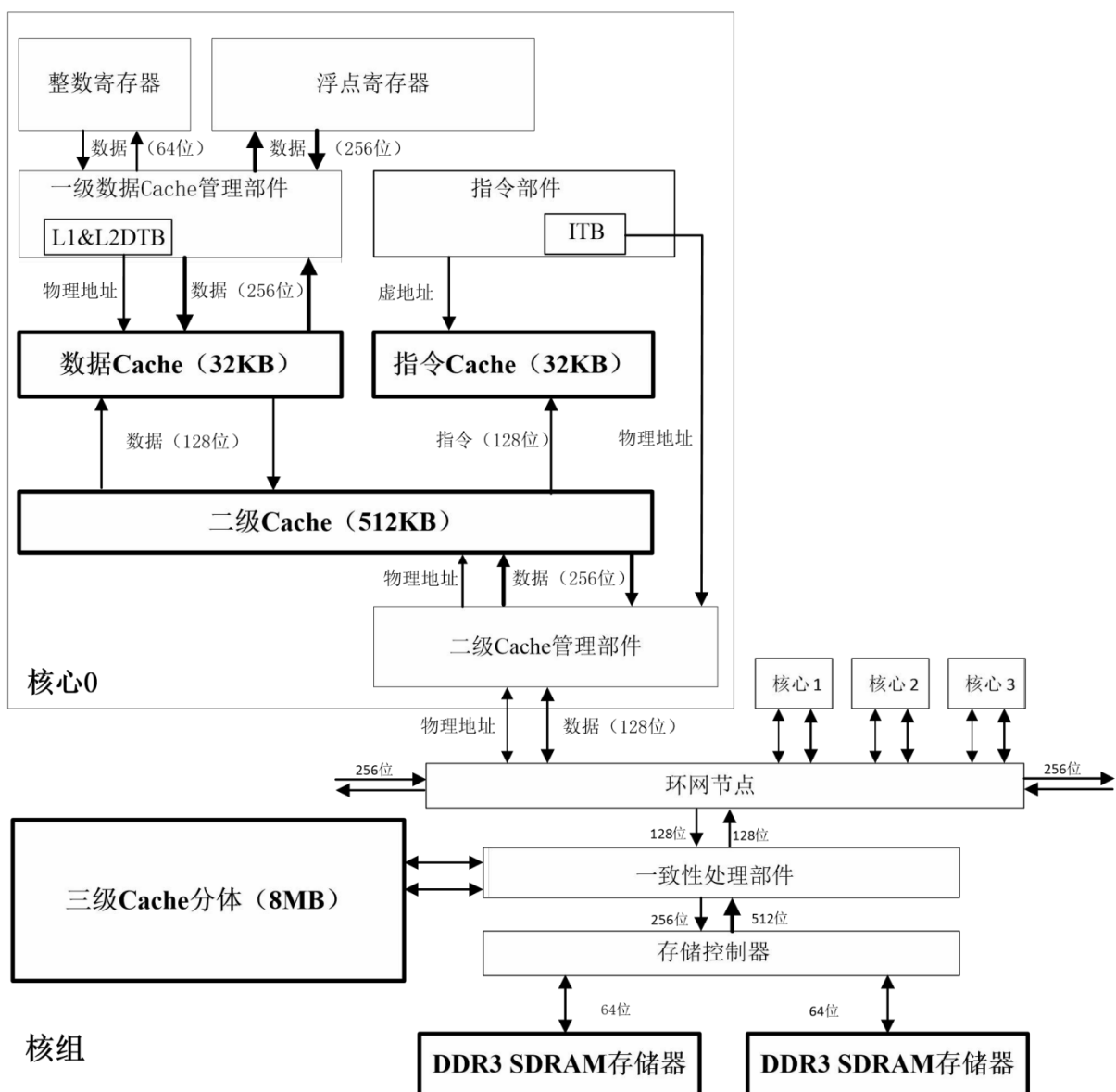


图 6-1: 申威 1621 处理器存储体系结构

5.1.1 寄存器

寄存器是层次化存储体系结构的最顶层，具有最小的容量，但具备最快的访问速度。申威 1621 核心包含整数寄存器和浮点寄存器，均采用多端口寄存器文件方式实现，其中整数寄存器文件由两套 64 位×105 寄存器组成，包含 5 个读端口和 5 个写端口；浮点寄存器文件为 64 位×95 条目×4

个体，每个个体均包含 7 个读端口和 4 个写端口。寄存器文件均支持旁路技术，实现对同一寄存器的同时写入和读出功能，以提高执行部件的性能。

整数寄存器文件与一级数据 Cache 之间的数据通路为 64 位×2，浮点寄存器文件与一级数据

Cache 之间的数据通路为 256 位，以支持 SIMD 短向量加速运算。

5.1.2 指令 Cache

申威 1621 处理器的每个核心都包含有一级指令 Cache。指令 Cache 容量为 32KB，Cache 行大小为 128B。指令 Cache 为 4 路组相联，采用虚地址索引和虚地址标记以及 FIFO 淘汰策略。

在物理实现上，指令 Cache 分为数据存储器阵列和标记存储器阵列，两者都采用偶校验方式来支持容错。在发生偶校验错时，指令 Cache 自动进行 Cache 刷新操作，同时在偶校验错使能时，还可向核心指令部件的中断和异常管理部件报告异常。

5.1.3 数据 Cache

申威 1621 处理器的每个核心都包含有一级数据 Cache。数据 Cache 容量为 32KB，Cache 行大小为 128B。数据 Cache 为 4 路组相联，采用虚地址索引和物理地址标记以及 FIFO 淘汰策略。

在物理实现上，数据 Cache 分为数据存储器阵列和标记存储器阵列，标记存储器阵列记录了相应 Cache 行的状态等信息。标记存储器阵列采用偶校验，可以通过软件方式纠正偶校验错。数据存储器阵列采用 ECC 校验，可以纠正单错、检测多错。

5.1.4 二级 Cache

申威 1621 处理器的每个核心都包含有二级 Cache。二级 Cache 容量为 512KB，Cache 行大小为 128B。二级 Cache 为 8 路组相联结构，采用物理地址索引和物理地址标记以及 FIFO 淘汰策略。

二级 Cache 既可以存放数据，也可以存放指令。二级 Cache 与一级数据 Cache 间为包含关

系，

即一级数据 Cache 内容保持为二级 Cache 的子集，硬件实现两级 Cache 之间的数据一致性。一级指

令 Cache 与二级 Cache 之间不存在包含关系，可通过配置选项设定硬件维持以及指令 Cache 的数据一致性，或者由软件解决一致性问题。

在物理实现上，二级 Cache 同样划分为数据存储器阵列和标记存储器阵列，并且都采用 ECC 校验，可以纠正单错、检测多错。

5.1.5 三级 Cache

申威 1621 处理器的 4 个核心集中共享三级 Cache。三级 Cache 容量为 6MB，Cache 行大小为 128B。三级 Cache 为 24 路组相联结构，采用物理地址索引和物理地址标记以及伪 LRU 淘汰策略。

三级 Cache 既可以存放数据，也可以存放指令。当配置为硬件支持指令 Cache 一致性时，三级 Cache 与 4 个核心内部的一、二级 Cache 间为包含关系，即一级数据 Cache、一级指令 Cache 和二级 Cache 内容保持为三级 Cache 的子集，硬件实现三级 Cache 与核心 Cache 之间的数据一致性；否则一级指令 Cache 与三级 Cache 之间不存在包含关系，指令 Cache 的数据一致性由软件解决。

在物理实现上，三级 Cache 同样划分为数据存储器阵列和标记存储器阵列，并且都采用 ECC 校验，可以纠正单错、检测多错。

5.1.6 主存

申威 1621 处理器片内集成了一路 DDR3 存储控制器。申威 1621 最大可支持 16GB 的主存容量。主存位于层次化存储结构的最底层，为处理器内部 4 个核心所共享。

基于申威 1621 处理器片的结构特点，每个核心对共享主存的访问也有两种方式，一种是共享 Cache 访问方式，即可在本核心的 Cache 中存放访问数据的副本；另一种是共享不可 Cache 访问方式，即不在本核心的 Cache 中存放共享数据的副本。

核心对主存空间的共享数据访问可以采用共享可 Cache 形式或共享不可 Cache 方式进行。

5.2 存储空间

5.2.1 虚空间

申威 1621 支持的虚拟地址为 64 位，实际实现 53 位（要求虚地址 VA[63:53]必须是虚地址 VA[52:0] 最高位的符号扩展），并引入了 2 位虚拟机号（VPN）和 8 位用户进程号（UPN），将

53 位的虚地址空间延伸到 63 位，其中 VPN 用于区分 4 个不同的虚拟机，UPN 用于区分同一虚拟机下的 256 个

不同的进程。

申威 1621 支持四种运行模式，即用户模式、内核模式、虚拟模式和硬件模式。不同模式下对虚地址空间有不同的访问限制，用户模式（UM）下只能访问 VA[52]为“0”的虚地址空间，且必须采用虚地址访问方式，不支持超页模式访问（一种将虚地址直接当作物理地址来访问的模式）。内核模式（KM）下只能访问 VA[52]为“1”的空间，且必须采用虚地址访问，也不支持超页模式访问。虚拟模式（VM）下可访问 VA[52]为“0”空间，且只能采用超页模式访问。硬件描述（HM）下可以访问整个虚地址空间，并同时支持虚地址和超页访问模式。虚拟模式与用户模式下访问的虚地址空间是重叠的，由操作系统进行管理。

5.2.2 物理空间

申威 1621 处理器支持 48 位物理地址空间，该空间按照物理地址最高位分为存储器空间和 I/O 空间，其中物理地址 PA[47]为“0”的物理空间为存储器空间。

在申威 1621 处理器四种运行模式中，只有在硬件模式下，指令流和数据流请求才可以使用物理地址访问物理空间，在其他三种处理器模式下都只能使用虚地址或者超页模式访问。而在层次化存储体系结构中，一级数据 Cache 和二级 Cache 都采用物理地址标记，因此在非硬件模式下访问数据 Cache 和二级 Cache，必须将虚地址转换为物理地址，才能正确访问。

5.3 存储管理

申威 1621 实现基于页面的存储管理方式。通过地址转换缓冲（TLB）实现虚页地址到物理地址的快速转换和存储保护。指令流和数据流使用不同的 TLB 进行虚实地址转换，即分别是指令流地址转换缓冲（ITB）和数据流地址转换缓冲（DTB）。

5.3.1 指令流地址转换缓冲

指令流地址转换缓冲（ITB）负责在指令 Cache 脱靶而需要访问二级 Cache 时，将指令流访问的虚地址转换为物理地址。

ITB 为全相联结构，共 32 个条目，支持 2 位虚拟机号和 8 位用户进程号。ITB 采用轮转替换策略，即装填 ITB 时，先选择无效的 ITB 条目，若不存在则选择替换淘汰指针所指向的 ITB 条目。核心支持 ITB 条目的装填、全部刷新或局部刷新。其中全部刷新是清除 ITB 中所有条目的有效位，而条件刷新只清除满足指定条件的 ITB 条目有效位。

5.3.2 数据流地址转换缓冲

数据流地址转换缓冲（DTB）负责将数据流访问的虚地址转换为物理地址，再访问一级数据

Cache。

DTB 采用两级结构，一级 DTB（L1DTB）为全相联结构，包含 64 条目，支持 2 位虚拟机号和 8 位用户进程号。L1DTB 采用轮转替换策略，即 DTB 装填时，首先选择无效的 DTB 条目；若不存 在则按轮转替换策略选择被淘汰的 DTB 条目。二级 DTB 包含 512 个条目、4 路组相联结构的基本

页面转换缓冲和 32 个条目、全相联结构的大页转换缓冲。当 L1DTB 脱靶时，申威 1621 会查询 L2DTB， 如果命中，则自动装填 L1DTB，否则产生相应的 DTB 脱靶异常。

对 DTB（包括 L1DTB 和 L2DTB）的刷新有五种方式，包括对指定虚地址的 DTB 条目刷新、对指定用户进程号的 DTB 条目刷新、对指定虚拟机号的 DTB 条目刷新、对指定虚拟机号的全部用户进程的 DTB 条目刷新和对全部 DTB 条目刷新。

5.4 一致性处理机制

申威 1621 处理器的存储空间采用分布共享存储（CC-NUMA）。由于核心存在两级 Cache，需要保证核心内两级 Cache 之间以及不同核心的 Cache 之间数据一致性。其中一级指令 Cache 的一致性由操作系统等软件维护，一级数据 Cache、二级 Cache 和主存中的共享数据一致性由申威 1621 处 理器的一致性处理机制来维护。

申威 1621 采用基于“三级 Cache 标记存储器”的一致性处理机制，通过“三方”写无效一致性 协议解决多 Cache 一致性问题，“三方”是指 Cache 一致性处理涉及到的请求源方（记为 local）、 请求目标方（记为 home）和数据副本拥有方（记为 owner）。这种一致性协议的特点有：

- 1) 核心 Cache（包括一级数据 Cache、一级指令 Cache 和二级 Cache）的内容为三级 Cache 的 子集；
- 2) 核心 Cache 支持 5 种数据副本状态：无效、清洁共享、清洁独占、脏共享和脏独占， 可支 持基于 MESI 或 MOESI 的多 Cache 一致性协议；
- 3) 核心 Cache 为每个数据块记录当前的状态，以进行 Cache 命中和写权限判断，在 Cache 不 命中或写权限禁止时，向下一级存储层次发出一致性访问请求；
- 4) 三级 Cache 标记存储器中记录了所有核心的 Cache 中数据块的分布与状态；标记存储

器中记录的 Cache 块状态分为 6 种：无效、清洁（核心无副本，三级 Cache 中的数据副本与主存数据一致）、脏（核心有副本，三级 Cache 中的数据副本与主存数据不一致）、清洁共

享（核心和三级 Cache 的数据副本与主存数据一致）、脏共享（核心和三级 Cache 的最新数据一致，但与主存数据不一致）、独占脏（唯一核心拥有脏副本），实现基于 MESI 的“三方”写无效一致性协议。

- 5) 在一致性请求处理中，核心只作为一致性事务的请求源方或数据副本拥有方，一致性处理部件（包括与之连接的三级 Cache 和存储控制器）只作为请求目标方。因此，一致性处理部件是一致性请求的串行点，保证存在冲突的请求按照顺序处理；
- 6) 通过“三方”写无效一致性协议，对读主存请求需要获得主存或者 Cache 中的最新副本，写主存请求需要在主存或者 Cache 中的最新副本上进行数据修改；
- 7) 对系统接口产生的主存访问请求，也通过标记存储器和“三方”写无效一致性协议，获得最新的主存数据或者在最新的主存数据上进行写修改；
- 8) 维护接口可以产生直接的存储器访问请求，不进行任何 Cache 一致性处理，主要用于维护调试等目的。

6 中断和异常

6.1 中断

中断是指来自于正常指令流之外、与指令流异步且需要核心进行干预和处理的事件。在申威 1621

处理器中，共 5 种类型的中断，并分别设置了 5 种不同的中断处理程序入口，按照处理优先级由高 到低排列，分别为：

- 1) 复位和睡眠唤醒中断；
- 2) 机器检查错中断；
- 3) 睡眠中断；
- 4) INT1：一般中断，包括核心内部中断和外部中断；
- 5) INT0：核间中断。

6.1.1 中断类型

1) 复位和睡眠唤醒中断

复位是一种特殊的中断，由申威 1621 引脚上的上电复位信号或冷复位信号通过电平方式触发。当这类信号有效时，对芯片进行完全复位，复位结束后，芯片进入初始化流程。

睡眠唤醒中断用于使处于睡眠状态的核心恢复正常运行。核心在硬件模式下可发出对其它核心的睡眠唤醒中断，维护命令也可以对指定核心发出睡眠唤醒中断。

2) 机器检查错中断

机器检查错是对申威 1621 处理器中各种严重错误的总称，机器检查错中断是在处理器内部发生 精确或非精确断点的严重故障时产生的中断。

3) 睡眠中断 睡眠中断的目的是使无工作负载的核心进入睡眠状态，从而降低芯片运行功耗。

核心在非用户

模式下可以向本核心或其它核心发出睡眠中断，维护命令也可以向指定核心发送睡眠中断。

4) INT1 中断

INT1 中断分为核心内部产生的已纠正错中断和定时器中断，以及核心外部产生的外部硬件中

断。

申威 1621 处理器具有对内部存储器和外部主存中出现的可纠正错进行预警功能，由核心的



寄存器或系统接口的控制寄存器中可纠正错预警阈值，当在指定时间里发生已纠正错的次数超过指定阈值时，触发已纠正错中断。

申威 1621 每个核心内部设置一个 64 位的计数器，计数器指定的初值开始，每个时钟周期进行减“1”计数，当计数器减到“0”时，便产生定时器中断。

外部硬件中断是来自核心外部的中断请求。申威 1621 处理器最多可以设置 523 个外部硬件中断，这些中断有来自 PCI-E 接口的一般中断（INTA/B/C/D）和 MSI 中断，也有来自接口处理模块的短时中断和来自维护接口的维护中断与故障中断。

5) INT0 中断

INT0 中断为核间中断。申威 1621 处理器的核心可以在任何模式下向一个或多个核心发出核间中断（包含自身），用于实现处理器核心之间的快速通信。核间中断可以细分为：核间中断 0/1，以及异步消息中断。

6.1.2 中断处理

核心内的各种中断类型中，已纠正错中断、定时器中断、机器检查错中断、睡眠中断和外部硬件中断采用向量中断方式，软件可通过核心内的 CSR 寄存器或外部相关 I/O 寄存器来指定每种中断的向量号。复位和睡眠唤醒中断采用非向量化管理。

申威 1621 处理器中，所有中断总是在指令边界上被识别，且保证中断断点之前的指令都无异常完成并退出。处理中断前，自动将断点处的指令地址保存到 CSR 寄存器中，以便中断处理结束后能正确返回到断点处继续执行，而后从相应的中断特权程序入口开始取指令执行，进行相应的中断处理。睡眠中断、睡眠唤醒中断和机器检查错中断各占用一个独立的中断特权程序入口，一般中断共用一个中断特权程序入口。从中断和异常管理部件接收到中断请求，到中断处理特权程序入口的指令进入核心流水线的译码站台，最快需要 10 个时钟周期。

6.2 异常

异常是指在指令处理过程中产生且与指令流同步的、由程序或软件原因而引起的自陷，申威 1621

处理器中的异常包括 3 种类型：

1) TLB 脱靶 (TLB Miss)

表示指令流或数据流地址进行虚实地址转换时发生 TLB 脱靶，具体又分为 DTB 用户脱靶、DTB 核心脱靶、DTB 二次脱靶、ITB 用户和 ITB 核心脱靶。为了便于软件快速确定 TLB 脱靶的类型，上述各种脱靶类型各自对应一个不同的异常处理程序入口。

2) 故障 (Fault) 表示指令非法或指令在处理过程中的操作非法，产生故障的指令被中止，而不是正常退出。故

障类异常包括浮点屏蔽故障、地址不对界故障、数据流故障、指令流故障、操作码非法。为了便于软件快速确定故障类型，上述各种故障类型各自对应一个不同的异常处理程序入口。

3) 算术自陷 (Arithmetic Trap) 表示运算类指令在运算时，操作数或运算结果出现异常，产生算术自陷的指令都将处理完毕并

正常退出。算术自陷包括整数溢出自陷、非精确结果自陷、下溢自陷、上溢自陷、除数为零自陷、无效操作自陷、原子操作溢出自陷。算术自陷设置有统一异常处理程序入口。

在申威 1621 处理器中，异常保留的断点都是精确的。发生异常时，指令携带异常标记并登记到核心指令部件的重排序缓冲中。当异常指令到达重排序缓冲的头部时，广播清空流水线，并在内部 CSR 寄存器中登记异常指令的地址、异常类型以及其它相关信息。从指令执行产生异常，到异常处理特权程序入口的指令进入译码站台，最快需要 9 个时钟周期。

7 低功耗设计与管理

申威 1621 处理器以高性能、低功耗作为设计目标，采用了多种低功耗设计技术和管理策略，降低芯片功耗或者运行时功耗，提高能效比。

7.1 低功耗设计

申威 1621 采用了包括门控时钟、多阈值晶体管设计、多频率设计和运行控制等多种低功耗设计技术。

7.1.1 门控时钟

申威 1621 处理器设计全面采用了门控时钟设计方法，在指令流水线、主要数据通路、Cache 存储器等设计中通过门控时钟的方式降低芯片的动态功耗和运行功耗。

7.1.2 多时钟设计

申威 1621 处理器内部根据性能平衡设计思想，以及不同组成部分的功能特点，采取了多时钟设计，将全芯片划分为多个不同频率的时钟域，具体如下：

- 1) 核心时钟域最高频率为 1.8GHz；
- 2) 一致性处理部件、核组间互连开关时钟域最高频率为 1.25GHz；
- 3) DDR3 存储器接口时钟域最高频率为 533MHz，从而实现 DDR3 存储器接口 2133Mbps 的数据传输率；
- 4) 存储控制器、PCI-E 接口、维护接口和测试接口，采用相对较低的多种频率的时钟。通过多时钟设计方法，可以在保证性能的前提下，减少高频率时钟的使用范围，从而降低芯片的动态功耗。

7.1.3 多阈值晶体管设计

申威 1621 处理器设计中，根据不同时钟域的时钟频率，在达到需要的设计目标前提下，综合使用两种阈值的晶体管，包括常规阈值（SVT）和高阈值（HVT）晶体管。对于非关键时序路径，采用 HVT 器件；对于时序关键路径，使用 SVT 器件，有效降低芯片的静态功耗（漏电功耗）。

7.1.4 运行控制

申威1621处理器设计中提供了不同层次的运行控制接口，通过这些接口可以控制内部不同门控的运行状态或运行速度，从而达到降低运行功耗的目的。

1) 核心或存储控制器断连

申威 1621 处理器可以在上电复位或冷复位时，可根据需要，通过维护接口配置寄存器，将 4 个核心中的任意核心，或存储控制器配置为断开状态。当申威 1621 处理器启动运行后，断开的核心处于很低频率的复位状态，没有断开的核心则处于正常工作状态。需要时可重新配置，打开处于断开状态的核心。

2) 核心深睡眠 通过向核心发送睡眠中断请求方式，可使核心进入“深睡眠”状态，此时核心，以及一致性处理部件中其对应的标记存储器处于很低频率的复位状态。需要恢复运行时，可通过唤醒中断来激活处于睡眠状态的核心，恢复正常工作状态。通过维护接口也可使得处于睡眠状态的核心恢复正常工作状态。

3) 核心浅睡眠 当核心执行停机指令（HALT）时，可使核心进入“浅睡眠”状态，此时核心的指令流水线将停止取指令，不再执行任何指令，也不访问 Cache 和存储器，但仍然支持 Cache 一致性操作，寄存器和 Cache 中的数据不会丢失。虽然浅睡眠状态下核心时钟仍然运行在工作频率下，但配合门控时钟，仍然可大大降低“浅睡眠”状态下核心的运行功耗。通过任何未屏蔽的中断可将处于此状态的核心“唤醒”，恢复正常工作状态。

4) 关闭浮点部件或 SIMD 部件

当软件或用户程序不使用浮点指令或 SIMD 扩展指令时，可通过 CSR 寄存器来关闭核心的浮点部件或 SIMD 处理部件的工作时钟，从而降低核心运行功耗。

5) 指令发射速度调整

在对核心的运行性能要求降低时，可通过 CSR 寄存器，降低核心发射指令的速度，从而减少执行部件和访存部件的信号翻转率，配合门控时钟，从而降低运行功耗。

6) 外部接口待机

申威 1621 处理器的 DDR3 存储器接口和 PCI-E 接口都可以自动或由软件配置而进入待机状态。在待机状态下，上述接口部件不产生任何数据传输，以降低接口功耗，直到产生新的接口访问请求，激活这些接口部件，开始数据传输。

7.2 功耗管理

申威1621提供了多种低功耗运行模式，软件可以根据需要，通过芯片提供的运行控制接口，协助芯片进入不同层次的低功耗运行模式。

7.2.1 关闭不使用的核心、存储控制器

申威 1621 处理器可以将不使用的核心或存储控制器配置为断开状态，从而达到关闭核心或存储控制器的目的。当申威 1621 启动运行后，断开的核心或存储控制器将一直处于很低频率的复位状态，降低不使用核心或存储控制器的动态功耗。

7.2.2 核心睡眠

当某个核心不需要运行时，可以通过向该核心发送睡眠中断，使其进入“深睡眠”状态，此状态下的核心将一直处于很低频率的复位状态，降低不工作核心的动态功耗。

当核心处于空闲状态时，软件可执行停机指令，使核心进入“浅睡眠”状态，此状态下的核心内部信号处于不翻转状态，降低空闲核心的动态功耗。

7.2.3 部件切割

当核心运行无浮点或无 SIMD 指令的应用课题时，软件可控制关闭浮点或 SIMD 部件，以节省运行功耗。

当软件不需要使用 DDR3 存储器接口或 PCI-E 接口时，可以在申威 1621 启动运行时，将上述指定接口设置为断开状态，或者通过软件配置来控制上述接口在空闲一段时间后，自动进入待机状态，以节省运行功耗。

7.2.4 核心调速

当核心运行较低工作负载时，软件可动态控制核心执行的效率，在满足基本需求的同时最大限度降低运行功耗。

8 可靠性设计

申威 1621 处理器采用 28nm CMOS 代工工艺实现，晶体管数量超过 3 亿，核心工作频率达到 1.8GHz。为了保证芯片的稳定可靠性，申威 1621 处理器采取了多项可靠性设计。

8.1 校验

申威 1621 处理器对芯片内的大容量存储器 and 主要数据通路都采用了校验技术，具体如下：

- 1) 一级指令 Cache：采用偶校验。由于指令的“只读”属性，出现校验错时可以通过 Cache 刷新进行纠错；
- 2) 一级数据 Cache：数据存储采用 ECC 校验，可实现纠单错并检测多错。标记存储采用偶校验，可通过软件配合对校验错进行纠错；
- 3) 二级 Cache：数据存储和标记存储都采用 ECC 检验，可实现纠单错并检测多错；
- 4) 二级 DTB：二级 DTB 中的 4 路组相联标准页面转换缓冲采用偶校验，出现校验错时，置无效对应的条目；
- 5) DDR3 存储器接口：采用 ECC 校验，可实现纠单错并检测多错；
- 6) 核心内部数据通路：采用 ECC 校验，可实现纠单错并检测多错；
- 7) 核组内部、系统接口互连数据通路：采用连续 ECC 校验方式，对传输过程请求和数据进行校验，数据部分纠单错并检查多错，请求部分检查单错和多错。

8.2 容错

申威 1621 实现了基于错误预警和部件隔离的容错策略。对于可纠正的错误，芯片内针对不同核心和外围模块分别设置可纠正错计数寄存器，当计数超过阈值时，可以通过中断向操作系统提出预警。对于可纠正错误发生次数过多的核心或外围模块，操作系统可以动态关闭相关核心或外围模块，并对正在运行的作业进行“降级”处理，将作业迁移到其它核心或处理器上运行。

当核心或外围模块发生不可纠正错误时，可通过维护系统将相关核心和外围模块切割掉，然后重新复位，在操作系统支持下，将课题“回卷”到上一个保留点，重新执行。

8.3 冗余

申威 1621 处理器中包含有大量存储器单元，所有核心的 Cache 总容量达到 2.25MB，因此其稳定可靠性直接与这些存储器的可靠性相关。

针对申威 1621 内部各类存储器的不同容量和特点，对存储器阵列进行分块控制，合理规划存储器的冗余方式和粒度，增加内部存储器阵列的自测试和冗余修复功能。

在申威 1621 处理器的加电和复位过程中，通过自测试寻找存储器阵列中的错误单元，并使用冗余单元对其进行自动替换、修复，保证芯片运行的可靠性。

8.4 电源控制

为保证申威 1621 处理器在加电或者对内部核心进行动态切割时，因功耗急剧变化而影响供电的稳定性，增加了电源控制功能，方法是：

- 1) 上电复位和冷复位结束后，系统接口、各存储控制器和 4 个核心的工作时钟依次切换到工作频率；
- 2) 芯片正常运行时进行冷复位，系统接口、存储控制器和 4 个核心的工作时钟依次从工作频率切换为很低的频率；
- 3) 处于“睡眠”状态的多个核心被“激活”时，这些核心将依次被唤醒，核心的工作时钟也依次切换到工作频率；
- 4) 处于“活动”状态的多个核心被系统切割或者关闭需要进入“睡眠”状态时，这些核心将依次从工作频率切换到很低的频率并处于复位状态，降低运行功耗。