

基于 ADL 描述的 ASIP 模型体系架构

朱 勇

(武汉纺织大学 数学与计算机学院, 湖北 武汉 430073)

摘 要: 为了获取 ASIP(专用指令集处理器)体系结构可执行模型,实现自动化设计,研究遵循“描述—综合”设计方法学,采取了 ADL(体系结构描述语言)手段,构建了对对象的“功能—结构—物理”三层描述,将复杂的 ASIP 系统划分为不同的模型视图. ADL 允许设计者以纯行为形式描述抽象复杂系统,且语言本身很容易被软件捕捉执行,因此 ADL 规约将驱动整个 ASIP 设计流程.

关键词: 体系结构描述语言;“功能—结构—物理”模型;专用指令集处理器;“描述—综合”设计方法学

中图分类号: TP33

文献标识码: A

文章编号: 1000-7180(2013)08-0053-04

The ASIP Model Based on ADL Description

ZHU Yong

(School of Mathematics and Computer, Wuhan Textile University, Wuhan 430073, China)

Abstract: The object “functional-structural-physical” three-layer description by ADL (Architecture Description Language) be built followed the “describe-synthesize” design methodology to acquire the ASIP (Application Specific Instruction Set Processor) executable model to realize automation design. And the complexity of the ASIP system is divided into different view of the model. ADL of pure forms of behavior is allowed to describe the abstract complex system. In addition, the language itself is very easy to be captured for implementation by software, so the ADL specification will drive all the ASIP design process.

Key words: ADL (Architecture Description Language); “Functional-Structural-Physical” Model; ASIP (Application Specific Instruction Set Processor); “Describe-Synthesize” Design Methodology

1 引言

ASIP 是介于 GPP(通用处理器)和 ASIC(专用集成芯片)之间的一种高性能处理器,很好地避免了 GPP 和 ASIC 各自的不足,解决了 GPP 在嵌入式系统中性能不高、功耗过大的问题,以及 ASIC 灵活性差、开发门槛高、产品风险大的缺点^[1]. ASIP 的这些优点使其被广泛应用于嵌入式系统中,特别是多媒体处理、网络信息处理和通讯领域. 随着嵌入式系统应用领域的不断扩大,ASIP 的这一技术特点使得其越来越受到嵌入式系统设计人员的青睐.

因处理器在信息领域中的重要地位,其核心技术一直是科技发达国家的重点,特别是一些国

际超级公司更是垄断了 GPP 产业链. 反观国内厂商的产品大多是采用国外主流 SoC 平台的“衍生品”,基于自主知识产权的 ASIP 核心寥寥无几. 与 GPP 和 ASIC 非常成熟的研究相比,ASIP 系统设计目前还没有通用的系统设计和开发工具,设计上更多的依靠系统设计人员的经验,很多问题还有待于进一步研究. 其中 ASIP 系统设计方法学便是亟待解决的问题. 基于这种状况,我国在重大科研计划中也给予“中国芯”极大的支持与关注.

文献[2]中提出的设计方法学正从“捕获—模拟”向“描述—综合”模式转变,后者的优势是允许设计者以纯行为形式描述系统,不包括实现细节,最终综合为系统实现. 这种方法能使系统结构用自动工

收稿日期: 2012-09-20; **修回日期:** 2012-11-13

基金项目: 国家自然科学基金项目“ASIP 行为逻辑及其综合的关键技术研究”(61142006)

具综合生成. 文献[3]中提出的 ASIP 描述语言 LISA(指令集语言)旨在建立关于可编程体系结构及其接口的规范描述,以填补面相结构的语言,如 HDL(硬件描述语言)和针对体系结构的指令语言间的距离. 目前 LISA 已成为 EDA 设计的重磅利器.

2 ADL 概述

模型仅仅是理论上的概念,还需要以一种具体的形式,如体系结构描述语言来捕获这些模型. ADL 可分为以下三种类型:

- (1)标准语言(standard language);
- (2)专用语言(application-specific language);
- (3)前端语言(fronted-end language)

目前,一些主流的 ADL 概述如下:

(1)UML/ XML

UML 作为第三代建模语言,严格地定义了对象元模型的语义,并为对象结构、行为的捕获和通信提供了符号表示. 其中,描述 ASIP 流水线状态机的状态图可用于捕捉系统行为;另外,UML 模型能自动生成可执行系统^[4],文献[5]中提出了一种设计模式将 UML 系统功能描述转换为 SystemC 可执行模式.

使用 XML 描述参数化的体系结构模型能够很好地匹配目标系统,开发者可以针对自己的需求定制标记结构,得到便于理解的描述实例,让设计具有扩展性. 现在已有很多的开源应用程序支持 XML 及 schema,使 UML/ XML 成为规范的设计文档标准.

(2)SystemC

最近, SystemC 设计语言(包括 SpecC、ArchC、HardwareC 此类的描述语言)在学界和业界引起了极大的兴趣. 由 OSCI 组织支持的 SystemC 已成为了实际的可执行系统设计标准,其中 SystemC TLM(Transaction Level Modeling)工作组已经制定了互联抽象标准. 文献[6]中使用 SystemC 用非侵入模式实现了一个 PSM 模型以及 JPEG 编码实例. SystemC 的许多高级构架有可能使其成为名副其实的系统级描述语言(System Level Design Language, SLDL).

(3)xADL

这类 ADL 更加注重针对处理器模型的描述,充分体现 ASIP 设计特色, LISA 便是其中的典型代

表. 它提供了存储、资源、指令集、行为活动、时序、微构架等组件模型的描述信息,用于产生软件工具及其 HDL 综合逻辑表示^[7].

国内研究者提出的 xpADL 由操作描述、指令描述、操作映射描述、结构部件描述、指令流水线和数据传输路径以及存储层次描述等部分组成,构成了对目标 ASIP 体系的行为和结构进行抽象和表示^[8].

论文借鉴 xADL 类架构模式,构建行为功能、体系结构、逻辑综合三层 ADL 模型描述^[9],容易为第三方 EDA 设计综合;继承 UML/XML 良好的语义及语法特性,与软件框架很好地融合,成为文档交换标准.

3 行为功能 ADL

ASIP 行为级 ADL 描述候选处理器指令集行为功能. 它首先为候选体系结构生成指令集模拟器,然后用该模拟器模拟执行应用程序,通过剖析模拟执行得到的应用程序特征以及设计者关心的处理器参数信息来指导指令集体系结构的优化,对优化后的指令集可以反复进行上述操作,直到设计出满足约束条件的指令集^[10].

行为级 ADL 设计的目标是要能准确、完备、方便地描述 ASIP 的行为级设计需要的所有信息. 为了对指令集进行行为级描述,行为级 ADL 需要对指令集体系结构的指令格式、指令编码、指令功能以及必要的硬件信息进行定义. 即要实现指令特性和功能的语义描述及其寻址方式和格式的语法定义.

综上所述,行为功能 ADL 描述的基本元素如下:

操作码:指令执行的功能代码. 为数据处理(算数和逻辑指令)、数据存储(寄存器和主存存储指令)、数据传输(I/O 指令)、控制(分支、调试等)类型.

源操作数引用:引用范围为寄存器、主存、I/O、立即数;寻址方式为立即寻址、直接寻址、间接寻址、寄存器寻址等.

结果操作数引用:引用范围同上;寻址方式同上.

由此得行为功能 ADL 基本域格式:

<func>	/行为功能 ADL 标识
<code>< /code >	/操作码描述
<src></src>	/源操作数引用描述

```

    <res></res>          /结果操作数引用
</func>
下面给出 AND 指令行为功能 ADL 描述:
<func>
    <code>
        <name>AND</name>      /指令名称
        <op>res = src1 & src2</op> /操作
    </code>
    <src>src1 ; reg ; src2 ; reg</src>
        /源操作数引用范围(寻址方式)
    <res>res ; reg</res>
        /结果操作数引用范围(寻址方式)
</func>

```

上述 ADL 以纯行为方式描述了 AND 指令的行为功能,遵循“描述—综合”设计方法学原则,与“捕获—仿真”设计模式相比有着更大的优势.指令执行的组织与实现将在体系结构 ADL 和逻辑综合 ADL 中完成.

4 体系结构 ADL

根据行为功能描述,体系结构应给出设计的实现视图,即执行指令功能的处理器构架.具体来说,体系结构 ADL 描述的基本元素如下:

(1)控制器:给出流水线(取指、译码、执行、存储器、写回)指令周期状态及其数据通道控制信号.

(2)寄存器

(3)ALU

(4)内部总线/系统总线

由此得体系结构 ADL 基本域格式:

```

<arch>          /体系结构 ADL 标识
    <ctr></ctr>    /控制器描述
    <reg></reg>    /寄存器描述
    <alu></alu>    /操作执行
    <bus></bus>    /内部总线(i)/外部总线(x)
</arch>

```

下面给出 AND 指令的流水线体系结构 ADL 描述:

```

<arch>
<state>
    <name>IF</name>      /取指阶段
    <ctr>enable_rom_oe</ctr>
    <reg>IR = rom_data</reg>
    <bus_x>rom_addr = PC</bus_x>
</state>
<state>

```

```

    <name>ID</name>      /译码阶段
    <ctr>op = @IR</ctr>
    <reg>src1 = @IR ; src2 = @IR ; res = @IR</reg>
</state>
<state>
    <name>EXE</name>      /执行阶段
    <reg>a = src1 ; b = src2 ; PC = PC+1</reg>
    <alu>c = @op</alu>
</state>
<state>
    <name>WB</name>      /写回阶段
    <reg>res = c</reg>
</state>
</arch>

```

对于分支行为,PC 的描述方式是 $PC = cond ? PC+1 ; res.$

需要指出的是,上述 ADL 描述中对某些控制和结构作了简化和融合,如相关的译码、析取操作作用 @ 表示,具体的内容与指令格式有关;寄存器赋值其实包含了总线及其数据流与控制信号的联合作用.

5 逻辑综合 ADL

逻辑综合 ADL 应给出上述指令行为及处理器结构的硬件实现形式,如逻辑图、布尔表达式、状态机、HDL 及其他第三方组件库,它们可以下载到 SoC 器件上执行.

主要要实现的部件有流水线状态机、操作译码、寄存器堆、ALU、总线、RAM(数据缓存、D-Cache)、ROM(I-Cache)等,分别描述如下:

(1)流水线状态机

流水线是执行指令的一种策略,大大地提高了处理器效率;状态机是实现的技术手段,状态的界定与指令阶段的划分相应,使得 CPU 的时钟频率最快.

(2)操作译码

将指令操作域译码为控制信号,一方面选择具体的 ALU 处理逻辑,一方面通过总线选择源、结果操作数引用.对于 RISC 架构处理器将会获得简洁有效的逻辑实现.典型译码逻辑描述如下:

```

ADD = /OP[5]. /OP[4]. /OP[3]. /OP[2].
      /OP[1]. /OP[0]. F1
SUB = /OP[5]. /OP[4]. /OP[3]. /OP[2].
      /OP[1]. /OP[0]. F2

```

AND = /OP[5]. /OP[4]. /OP[3]. /OP
[2]. /OP[1]. /OP[0]. F3

...

(3) 寄存器堆

其核心部分是一组寄存器,接口模式要满足指令格式要求. 典型的形式为两读一写三个端口:读操作时给出需要读出的寄存器索引,然后输出其内容;写操作中给出写入寄存器的寄存器号和数据,接着把该数据写入.

(4) ALU

它是完成指令数据处理的具体所在,按照操作译码信号选择相应的处理单元,执行算术逻辑操作. 由于这些处理单元都有一些成熟的逻辑设计参考,不再赘述.

(5) 总线(内部总线、系统总线)

总线属于标准逻辑组件,可用三态门等方式实现. 关键是要根据控制信号实现相应的数据流,一般采用多路选择器逻辑.

(6) RAM、ROM

对于这种标准存储器件可直接调用综合好的组件,如 Quartus 软件中的 LPM 模块(大多数 EDA 都提供此类形式的调用).

将上述部件描述封装为类库形式,以便于 EDA 软件调用,实现可执行设计规约.

6 结束语

从上述描述可以看出,ADL 具有抽象、无二义的特性,非常适合 ASIP 模型描述;同时它的一致性保证了模型各层间的映射,为实现“描述—综合”设计方法奠定基础;另外,由于语言本身很容易被软件捕捉执行,因此 ADL 规约将驱动整个 ASIP 设计流程.

将复杂的 ASIP 系统划分为不同的模型视图使得设计具有可操作性,采用各自合适的技术来实现相应的模型,再通过接口将它们关联起来.

需要进一步解决和关注问题:

(1)各模型间映射. 采用目前得到广泛认可的 XML 语法与数据结构,保证了基础对象语义与接口的一致性.

(2)可执行规约. 借助语言可被捕捉执行的基本

特性,驱动整个设计流程.

(3)可扩展的体系架构. 适应多种 ASIP、RISC、TTA、MCU 构架.

参考文献:

- [1] Tilman Glokler, Heinrich Meyr. Design of energy-efficient application-specific instruction set processors [M]. USA: Springer Science Business Media, 2004.
- [2] Daniel D Gajski. Specification and design of embedded systems [M]. 边计年, 译. 北京: 机械工业出版社, 2005.
- [3] Oliver Schliebusch. Architecture implementation using the machine description language LISA [C]//Proceedings of the 15th International Conference on VLSI Design. India: Bangalore, 2002: 239 - 244
- [4] Bruce Powell Douglass. Developing real-time system with UML, objects, frameworks, and patterns [M]. 柳翔, 译. 北京: 机械工业出版社, 2005.
- [5] Joaquín Pérez. System behaviour capture: from UML to systemC [C]//forum on specification and design languages. Germany: Stuttgart, 2008.
- [6] Kim Gröutner, Wolfgang Nebel. Modeling program-state machines in systemCTM [C]//Forum on Specification and Design Languages. Germany: Stuttgart, 2008.
- [7] Miguel Angel. Designing highly parameterized hardware using xHDL [C]//Forum on Specification and Design Languages. Germany: Stuttgart, 2008.
- [8] 杨君. 专用指令集处理器(ASIP)体系结构设计研究 [D]. 合肥: 中国科技大学, 2006
- [9] Yong Zhu. ASIP specification based on multi-level ADL [C]//The 1st International Workshop on Electronic Equipment and Manufacturing Technology. China: Henan, 2010: 4157 - 4160
- [10] 陈艾. 专用指令集处理器(ASIP)行为级设计方法研究 [J]. 计算机工程与应用, 2004(29): 44 - 46.

作者简介:

朱勇男, (1964—), 博士, 教授. 研究方向为 ASIP 设计方法学、嵌入式系统及 SoC、实时图像处理.

