文章编号:1671-4598(2016)06-0315-04 DOI:10.16526/j.cnki.11—4762/tp.2016.06.086 中图分类号:TP301 文献标识码:A

机器人控制系统关键模块的形式化验证

娄晨辉，李晓娟，关 永
（首都师范大学 高可靠嵌入式系统技术北京市工程研究中心 电子系统可靠性重点实验室 轻型工业机器人与安全验证实验室，北京 100048）

摘要：随着机器人应用在越来越多的领域，人们对其安全性的要求越来越高，作为机器人的核心，控制系统设计的可靠性对整个系统的安全至关重要；针对一种模块化设计的机器人控制系统架构，利用 xMAS（eXecutable MicroArchitecture Specification，可执行微架构描述）模型在定理证明器 ACL2 中对其功能正确性进行验证，首先对 Xmas 在 ACL2 中的形式化理论做了阐述，然后对该机器人控制系统中的加速度传感器数据采集模块建立 xMAS 模型，提取关键属性并进行验证；将 xMAS 模型和定理证明器 ACL2 相结合，可以很好地解决机器人控制系统的验证问题，为机器人控制系统的形式化验证提供一个有效的方法参考。

关键词：机器人控制系统；传感器数据采集模块；形式化验证；定理证明；微架构模型

Formal Verification of Key Module in Robot Control System

Lou Chenhui, Li Xiaojuan, Guan Yong
(Highly Reliable Embedded Systems Lab, Reliability Key Laboratory of Electronic Systems, Capital Normal University, Light Industrial Robot and Secure Verification Laboratory, Beijing 100048, China)

Abstract: As robots used in more and more fields, people are more stricted with their safety. As the core of the mobile robot, the reliability of the control system is very important to the whole system. In this paper, a modular design of robot control system architecture is modeled by the xMAS and then verified using ACL2, proving the funtionality correctness. As the formalization of xMAS model in acl2 is not completed, we first improve the formalization process in acl2 and then establish xMAS model of the sensor data acquisition module, abstract some key properties and then verify them. We combine the theorem prover ACL2 and xMAS model, which is a great way to solve the verification problem of robot control system, could also provide an effective reference method for the correctness verification of robot control system.

Keywords: robot control system; sensor data acquisition module; formal verification; ACL2; xMAS

0 引言

基于 ARM 和 FPAG 的机器人控制系统架构[1-2]，由于其模块化方便、功耗低、成本低廉、体积小和良好的扩展性等优点，逐步成为中小型移动机器人的通用性设计架构，其应用包括足球机器人[3]、救火机器人等等。机器人控制系统为移动机器人提供基础性的工作平台，其设计的可靠性对机器人应用的安全至关重要，软件系统故障可能导致机器人自毁或伤人事故。因此，对控制系统进行验证具有很大的现实意义。

结合控制系统设计模块化和分层控制的发展趋势，文献[4]设计了一种可用于多种移动机器人的模块化控制系统，可根据机器人应用的不同环境进行相应的定制。

在文献[4]中，用仿真的方法对机器人控制系统的正确性进行验证，其结果表明该控制系统可以满足设计需求，但由于仿真的不完备性，其结果是不可靠的，并不能完全保证控制系统的可靠性。形式化验证是运用数理逻辑来搭建系统的数学模型，证明必要的属性，从而验证设计的正确性，可以作为可靠地方法来验证控制系统。定理证明是形式化验证的一种，其基本思想是用函数来表示系统的行为特征，用定理来表示系统的性质。

本文提出采用定理证明的方法对机器人控制系统中的关键模块进行验证，所做工作基于一阶定理证明器 ACL2[5]，在保证控制系统可靠性的同时，引入一种优秀的建模及验证方法，可以方便的基于该工作对机器人控制系统的其他模块进行验证。

本文第一部分对 xMAS 模型进行简单的介绍；第二部分讲述了控制系统中的加速度传感器数据采集模块；第三部分为 xMAS 在 ACL2 中的形式化工作；第四部分包括对加速度传感器模块的形式化建模，提取属性并验证；第五部分总结全文，并给出未来的工作重点。

1 xMAS 模型

xMAS[6]模型是由 Intel 公司提出的一套图形化的建模工具，可用于设计和验证通信架构，它由 8 个基本元件组成，每个元件都有确切的语义，图 1 是基本元件图。

xMAS 元件的语义是通过布尔等式进行定义的，由于篇幅所限，本文只对元件的语义进行概括性的介绍，详细定义参照文献[6]。Fork 元件用一个分组作为输入，产生两个分组，

收稿日期:2016-03-17； **修回日期**:2016-04-12。
基金项目:国家自然科学基金项目(61373034);北京市自然科学基金(4122017)。

作者简介:娄晨辉(1992-),男,河南新乡市人,硕士研究生,主要从事形式化验证方向的研究。
李晓娟(1968-),女,内蒙古人,博士,教授,硕士研究生导师,主要从事形式化验证,计算机网络,机器学习方向的研究。
关 永(1966-),男,内蒙古人,博士,教授,博士研究生导师,主要从事形式化验证,电子系统健康状态预测,高可靠嵌入式系统与智能信息处理方向的研究。

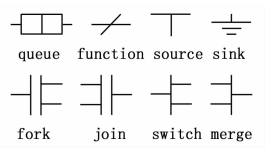

图 1 xMAS 基本元件图

且当且仅当一个入口和两个出口同时处于就绪状态的时候才会产生数据流动，join 元件则完成相反的操作。Merge 元件对两个输入的数据做出仲裁，可以根据需要选择合适的仲裁策略来选择发送的数据。Switch 元件完成路由的功能，根据分组的类型选择转发的通道。Source 和 sink 分别为数据的生产者和消费者，且上述两个元件为不确定性的。Function 元件是计算模块，可以对分组的数据或者控制部分做相应的改变。Queue 元件是唯一可以存储分组的元件，是标准的先进先出缓冲序列。元件之间用通道进行连接，通道有 3 种类型的信号：irdy、trdy 和 data，上述 3 个信号是一致的，若 irdy 为真，则它会一直保持真的状态直到 trdy 为真，然后进行数据的传输。另外，需要注意的是，元件和通道都是有类型的。

对于待验证的系统，首先要将其编码为单时钟的 xMAS 同步网络，然后对该网络提取相应的属性进行验证。

2 加速度传感器数据采集模块

在机器人控制系统中，ARM 微处理器为中央控制中心，实现复杂的上层控制算法及与上位机进行交互，FPGA 作为协处理器，负责外部传感器信息和电机驱动器的控制。在本文中，我们针对与 FPGA 直接相连的加速度传感器数据采集系统进行验证，传感器模块负责采集周期脉冲的占空比，图 2 为周期脉冲的状态转移图。

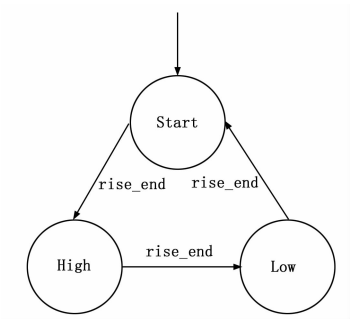

图 2 串口模块结构图

加速度传感器模块中用到同步状态机的概念。同步状态机是一组状态寄存器组成，来记忆状态机当前所处的状态。如果状态寄存器是 n 位的，那么该状态机最多可以记忆 2^n 个状态，使用共同时钟来控制状态改变的时刻，状态是否改变将取决于产生下一状态的组合逻辑 F，F 是由当前状态和输入信号的函数。状态机的输出是由输出组合逻辑 G 提供的，G 也是由当前状态和输入信号的函数。同步状态机用于存储当前的状态，并根据激励信号来改变自身的状态，向传感器模块发送下一脉冲时的状态。以下为有限状态机风格的周期脉冲占空比采集程序算法。

```
if reset=0
    cycle_high=0;cycle_low=0;duty_reg=0;state=0;
else
    if state='Start'
      if (rise_end)
          cycle_high=1;state=2'b01;
    else
    cycle_high=0;
if state='High'
    if (fall_end)
cycle_low=1;state=2'b10;
    else
cycle_high=cycle_high+1;
if state='Low'
    if (rise_end)
       duty_reg=cycle_high+cycle_low;
       cycle_high=cycle_high*100;
    duty_reg=cycle_high/duty_reg;
         cycle_low=1;state=2'b00;
    else
    cycle_low=cycle_low+1;
else
    state=2'b00
    end
```

3 xMAS 的形式化理论

xMAS 是高阶微架构模型，由于其抽象程度较高，可以有效减少系统的状态空间。目前，xMAS 模型已经用于大型通信系统的死锁检测、提取不变式[6]等等，但由于模型检测技术的局限性，在遇见超大型网络时会出现状态爆炸的问题，因此我们考虑将 xMAS 模型与定理证明技术相结合，即在一阶定理证明器 ACL2 中进行形式化，定理证明技术的原理是将属性表示为一个数学公式，利用数学规则进行推导，因此不会出现状态爆炸的问题。

目前关于 xMAS 模型在 ACL2 中进行形式化的问题所做的相关研究较少，文献［7］在这方面做了一定的工作，但由于其形式化策略对 GeNoC 平台的依赖，只对 xMAS 模型中较简单的 5 个元件进行了形式化：source、sink、switch、function、queue，这对 xMAS 模型的应用造成了一定的局限性，本文对 xMAS 在 ACL2 中的形式化过程进行改进，重点添加对剩下的 3 种元件的形式化描述。

首先，用 defstructure 宏定义一种结构来分别表示元件和通道。

```
(defstructure component type ins outs field);        (1)
(defstructure channel init target);                  (2)
```

式（1）和式（2）分别定义了元件和通道，元件有 4 个域，type 表示类型，ins 表示输入通道，outs 表示输出通道，field 表示元件的函数。通道有两个域，分别表示初始元件和目标元件。

在 ACL2 中，上述三种元件的 field 通过 alist 来表示，形如’(‘(* * *)‘(* * *))，对于 merge 和 join 元件，每个子序列表示两个参数对应一个相应的函数值，每种元件都有对应的函数从 field 域中得到其函数值，代码定义如下所示。

函数 xmas－transfer－calculate[7] 计算 xMAS 网络中数据流的流动，它有 5 个参数：

xmas－transfer－calculate (flg channel ntk unvisited ntk-state)

其中 flg 为要计算的信号值：irdy、trdy 和 data；channel 为当前遍历通道，ntk 为当前网络，ntkstate 为当前网络状态，unvisited 记录当前网络中未被计算的通道与信号的组合，其初始值为所有的通道和信号的组合，用该变量的值来保证函数的终止性；它的主要结构是根据当前信号值和通道进行计算，对整个网络进行遍历。为了将上述三个元件添加到其中，需要在该函数中添加新的分支选择结构，其中 irdy 与上述三种信号对应的代码如下所示：

```
((equal flg 'irdy)
(let* ((cpt (get-init-component channel ntk))
(type (component-type cpt))
(index-in (if (equal (get-in-channel cpt 0 ntk) channel) 0 1))
(next-unvisited (remove1 (cons channel flg) unvisited)))
(cond
((equal type 'merge)
(or
(xmas-transfer-calculate 'irdy (get-in-channel cpt 0 ntk) ntk next-unvisited ntkstate)
(xmas-transfer-calculate 'irdy (get-in-channel cpt 1 ntk) ntk next-unvisited ntkstate)))
((equal type 'join)
(and
(xmas-transfer-calculate 'irdy (get-in-channel cpt 0 ntk) ntk next-unvisited ntkstate)
(xmas-transfer-calculate 'irdy (get-in-channel cpt 1 ntk) ntk next-unvisited ntkstate)))
((equal type 'fork)
(and
(xmas-transfer-calculate 'irdy (get-in-channel cpt 0 ntk) ntk next-unvisited ntkstate)
(xmas-transfer-calculate 'irdy (get-out-channel cpt index-out ntk) ntk next-unvisited ntkstate))))))
(defun apply-field-join-and-merge (component param)
(let ((field (component-field component)))
(if (endp field)
nil
(if (xmas-join-equal (car field) param)
(cadr field)
(apply-field-join (cdr field) param)))))
(defun xmas-join-equal (field-item param)
(if (and (equal (nth 0 field-item)
(nth 0 param))
(equal (nth 1 field-item) (nth 1 param)))
t
nil))
```

4　验证和分析

为了对加速度传感器模块的功能正确性进行验证，首先要先对其建立形式化模型[8-9]。在传感器模块的工作过程中，同步状态机根据脉冲来决定下一周期的激励信号，结合 xMAS 元件的语义和加速度传感器模块的功能，分别给出同步状态机和加速度传感器模块的 xMAS 模型图，如图 3 所示。

在上述模型中，src0 模拟 ARM 向加速度传感器模块发送占空比，经过 func0 元件的处理将其转换为帧数据结构，然后经过 switch0 元件分发至不同的模块进行处理。如果 reset 信号为真，则通过通道 4 进入 join1 元件，结果经过 que3 元件的缓存后，进入 sink0 元件，结束该周期的处理；如果 reset 信号为假，即进入正常的处理周期，数据帧经过 switch0 元件，转发至 fork0 元件，fork0 元件根据数据帧结构的内容进行相应处理，如果当前为高脉冲，则高脉冲计数器 que1 加 1，如果为低脉冲，则低脉冲计数器 que2 加 1，处理完帧数据后，que1 和 que2 中的值由 join0 元件计算当前周期的占空比，然后发送至 join1 元件，经过 que3 元件缓存之后，进入 sink0，完成本周期的处理。

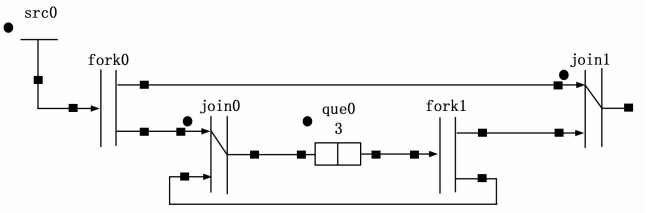

图 3　同步状态机 xMAS 模型图

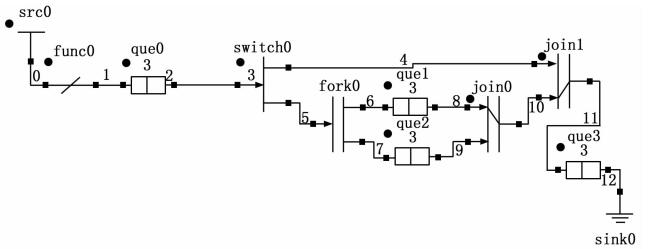

图 4　加速度传感器模块 xMAS 模型图

在建立 xMAS 模型之后，需要对其中的元件和通道进行定义，这也是对 xMAS 网络进行验证的最关键的一步。根据第四部分中元件和通道的形式化描述，对上述 xMAS 模型的元件和通道进行定义。

```
(defconst *func0* (component 'function '(0) '(1) '((num (sensor-frame num)))))
(defun sensor-frame (num)
(if (is-legal num)
(cons (high-pulse-generator num) (high-pulse-generator num))
nil))
(defun high-pulse-generator (num)
(if (> num 0)
(list 1 (high-pulse-generator (- num 1)))))
(defun low-pulse-generator (num)
(if (> num 0)
(list 0 (high-pulse-generator (- num 1)))))
(defconst *que0* (component 'queue '(1) '(2) '()))
(defconst *switch0* (component 'switch '(3) '(4 5) '('(frame-data (switch-transfer frame-data)))))
(defconst *fork0* (component 'fork '(5) '(6 7) '('(frame-data (high-low-pulse-transfer frame-data)))))
(defconst *que1* (component 'queue '(6) '(8) '()))
(defconst *que2* (component 'queue '(7) '(9) '()))
(defconst *join0* (component 'join0 '(8 9) '(10) '('(num1 num2 (calculate-duty-cycle num1 num2)))
(defun calculate-duty-cycle (num1 num2)
```

```
(* 100 (/ num1 (+ num1 num2))))
```

如上是元件在 ACL2 中的定义代码，其中 defconst 是 ACL2 中用于常量定义的函数，将元件定义为常量方便后续对其的引用，在 join2 元件中，通过将当前输入的时钟变量进行循环赋值来模拟分频操作。下面给出部分通道的定义，每个通道连接两个元件。

```
(defconst 'channel0' *src0* *func0*)
(defconst 'channel1' *func0* *que0*)
(defconst 'channel2' *que0* *switch0*)
(defconst 'channel3' *switch0* *merge0*)
(defconst 'channel4' *switch0* *fork0*)
(defconst 'channel5' *fork0* *que1*)
```

在机器人的工作过程中，加速度传感器数据采集模块可以检测移动机器人相对地面的倾斜角，获取移动机器人的姿态，对移动机器人的爬坡、避障至关重要，该模块的输出为周期可调的脉宽调制信号，其占空比与它感受到的加速度成正比，因此，正确的采集到占空比是至关重要的一个环节，本文针对上文建立的 xMAS 模型，提取以下两个关键属性[10]并在 ACL2 中进行验证。

属性 1：消息一致性，即由 FPGA 发出的占空比信号，经过 xMAS 网络的传递，最终能够采集到正确的占空比。ACL2 代码实现如图 9 所示。上述代码第五行用到的函数表示从 src0 注入网络的消息，最终会从 sink0 排出网络，且消息一致。在上述定理的限制下，能够保证消息发送的一致性。

```
(defthm sensor-duty-cycle-formal-collect
(let ((result (xmas-transfer-calculate 'data channel ntk unvisited ntkstate)))
(implies (and (xmasnetworkp ntk)
(member-equal channel (xmasnetwork-channels ntk))
(sensor-invariant src0))
(equal (arm-src-generator type)
(sensor-module-invariant sink0)
(defthm xmas-transfer-implies-available-space
(let (result (xmas-transfer-calculate 'trdy channel ntk unvisited ntkstate))
(implies (and (xmasnetworkp ntk)
(member-equal channel (xmasnetwork-channels ntk)))
(xmas-can-receive resource ntkstate))))
```

属性 2：队列中有足够的空间接收即将发送的数据包。在消息的传递过程中，会经过多次转发，在队列中进行暂存，该定理保证了每次转发的目的队列有足够的空间来接收当前消息。

上述所有的代码都在 ACL2 中运行通过，图 5 和图 6 分别给出属性 1 和属性 2 在 ACL2 中的运行结果，表 1 给出了属性验证的主要数据，时间测算基于因特尔酷睿 i5-3230M 处理器。

表 1 验证数据

	代码行数	验证步数	验证时间/s
属性 1	7	4 654	0.28
属性 2	5	1 357	0.18

5 总结

本文首次将 xMAS 模型在 ACL2 中进行完全形式化，并

```
Q.E.D.

Summary
Form:  ( DEFTHM SENSOR-DUTY-CYCLE-FORMAL-COLLECT ...)
Rules: ((COMPOUND-RECOGNIZER NATP-COMPOUND-RECOGNIZER)
                (:DEFINITION XMASNETWORKP)
                (:DEFINITION MEMBER-EQUAL)
                (:FAKE-RUNE-FOR-TYPE-SET NIL)
                (:REWRITE CAR-CONS)
                (:REWRITE CDR-CONS)
                (:TYPE-PRESCRIPTION MEMBER-EQUAL))
TIME:  0.28 seconds (prove: 0.28, print: 0.00, other: 0.00)
Prover steps counted:  4654
SENSOR-DUTY-CYCLE-FORMAL-COLLECT
```

图 5 属性 1 的验证结果

```
Q.E.D.

Summary
Form:  (DEFTHM XMAS-TRANSFER-IMPLIES-AVAILABLE-SPACE ...)
Rules: ((:DEFINITION XMASP)
        (:DEFINITION XMAS-NETWORKP)
        (:EXECUTABLE-COUNTPART CONSP)
        (:EXECUTABLE-COUNTPART XMAS-CAN-RECEIVE)
        (:REWRITE CDR-CONS)
TIME:  0.18 seconds (prove: 0.18, print: 0.00, other: 0.00)
Prover steps counted:  1357
XMAS-TRANSFER-IMPLIES-AVAILABLE-SPACE
```

图 6 属性 2 的验证结果

利用该模型对机器人控制系统中的加速度传感器数据采集模块的功能正确性进行验证。本文的验证结果可以保证机器人控制系统加速度传感器数据采集模块的可靠性，同时该方法也为机器人控制系统关键模块的验证提供了新的思路。在以后的工作中会基于 xMAS 对异步通信进行研究，使其可以在异步通信领域发挥其长处。

参考文献：

[1] 龚建伟. 地面机器人控制系统的研制 [J]. 计算机测量与控制，2003，11 (1)：36-38.

[2] 李 莉，吴星明，陈伟海. 基于 ARM 和 FPGA 的机器人运动控制器的实现 [J]. 计算机测量与控制，2007，15 (9)：1172-1173.

[3] 柳 林，郑志强. 基于 87C196KC 的足球机器人控制系统设计 [J]. 计算机测量与控制，2002，10 (7)：449-451.

[4] 陈剑斌. 基于 ARM 与 FPGA 的移动机器人控制系统设计 [D]. 广州：华南理工大学，2011.

[5] ACL2 证明 [EB/OL]. http：//zh. wikipedia. org/wiki/ACL2.

[6] Chatterjee S，Kishinevsky M. Automatic generation of inductive invariants from high-level microarchitectural models of communication fabrics [J]. Formal Methods in System Design，2012，40 (2)：147-169.

[7] Gastel B V，Schmaltz J. A formalisation of XMAS [J]. Eprint Arxiv，2013.

[8] Chatterjee S，Kishinevsky M，Ogras U Y. Quick formal modeling of communication fabrics to enable verification [A]. High Level Design Validation and Test Workshop (HLDVT)，2010 IEEE International [C]. IEEE，2010：42-49.

[9] Verbeek F，Schmaltz J. Formal specification of networks-on-chips：deadlock and evacuation [A]. Proceedings of the Conference on Design，Automation and Test in Europe. European Design and Automation Association [C]. 2010：1701-1706.

[10] Gao Y，Li X J，Guan Yong，et，al. Theorem Prover ACL2-based Verification of node communication in the robot operating system ROS [J]. Journal of Chinese Computer System，2014，35 (9)：2126-213.