

基于 SOPC 和量子框架的电控机械式变速器电控系统*

綦声波^{1,2} 张承瑞¹ 罗映¹

(1. 山东大学机械工程学院, 济南 250061; 2. 中国海洋大学工程学院, 青岛 266100)

【摘要】 针对传统嵌入式系统开发中软/硬件分离以及软件开放性不足等问题, 提出利用软/硬件协同设计和量子框架的综合解决方案, 形成了“五层/三总线/一框架”的软/硬件综合开发结构, 并将其应用于 AMT 电控系统的设计中。实现了针对 Nios II 软核的量子框架移植, 并设计了 AMT 主活动对象的层次状态机。实现过程及试验结果表明, 此种设计方案可提高系统开发的灵活性和软件的质量。

关键词: 电控机械式变速器 嵌入式系统 协同设计 量子框架 层次状态机

中图分类号: TP23; U463.6 **文献标识码:** A **文章编号:** 1000-1298(2011)10-0013-07

AMT Electronic Control System Based on SOPC and Quantum Framework

Qi Shengbo^{1,2} Zhang Chengrui¹ Luo Ying¹

(1. School of Mechanical Engineering, Shandong University, Jinan 250061, China

2. College of Engineering, Ocean University of China, Qingdao 266100, China)

Abstract

Separation of software development and hardware development, and the lack of open in software, are all shortcomings of traditional embedded system. A comprehensive solution using software/hardware co-design and the quantum framework was presented and applied to AMT electrical control system design. Thus, a comprehensive development framework of “five layers/three buses/one framework” was formed. The embedded vanilla kernel was selected, and the quantum framework port about vanilla kernel and Nios II was given, as well as the designed main active object’s HSM, which was more suitable than finite state machine for AMT system.

Key words Automated mechanical transmission, Embedded system, Co-design, Quantum framework, Hierarchical state machine

引言

传统的嵌入式系统开发, 软件和硬件往往独立进行, 存在一次性工程费用大、设计修改难, 研制周期不能有效保证等缺点。传统的软件架构, 软件组件之间的依赖性较大, 开放性不足, 从而导致软件升级困难、复用性差。改变传统嵌入式系统的开发模式, 提高软件的开放性和复用性, 成为目前汽车电子发展过程中亟待解决的问题。

汽车自动变速控制系统为典型的嵌入式汽车电子系统。文献[1]研究了自动换挡控制系统的硬件

和软件, 采用了 80C196KC 硬件平台和模块化的软件设计思想。文献[2]以重型卡车 AMT 为例, 设计了以 DSP 芯片 TMS320LF2407A 为核心的硬件控制系统, 并阐述了应用量子框架技术开发嵌入式系统的方法与步骤。文献[3]提出了用量子框架来构建开放式汽车电控系统体系结构思路。文献[1~2]都采用了软/硬件分离设计的设计思想, 文献[3]则是一种总体的软件开放性架构思路, 未涉及硬件结构。

本文提出一种基于 SOPC 的软/硬件协同设计及量子框架的“五层/三总线/一框架”的开放性总

收稿日期: 2011-03-07 修回日期: 2011-05-20

* 国家自然科学基金资助项目(50575129, 51075241)

作者简介: 綦声波, 博士生, 中国海洋大学副教授, 主要从事嵌入式技术及汽车电子技术研究, E-mail: qishengbo@ouc.edu.cn

通讯作者: 张承瑞, 教授, 博士生导师, 主要从事数控及汽车电子技术研究, E-mail: zhangchengrui@gmail.com

体系统结构,并应用于电控机械式变速器(AMT)电控系统的开发中。

1 AMT的软/硬件协同设计

SOPC(system on programmable chip)是基于大规模FPGA的单片系统,它将系统构建的灵活性和自主权留给用户,可以在单芯片上实现单个或多个“微处理器核”及各种IP构成的应用系统,既稳定可靠、实时性强,又设计灵活多样、可变性强,可以在设计过程中协同硬件子系统和软件子系统,充分利用设计的空间维度和时间维度,是现代嵌入式系统设计的必然趋势和重要途径。

AMT换挡控制单元TCU是整个电控系统的核心,其设计思路及性能优劣往往决定了AMT的主要性能指标^[4]。在TCU的SOPC设计中,软硬件的界限开始模糊,若干功能模块既可以由硬件实现,又可以由软件实现,因此需要综合分析系统软硬件功能及现有资源,最大限度挖掘系统软/硬件之间的并行性,协同设计软硬件体系结构,并在协同设计过程中,不断进行软硬件任务的重新划分和修改,以使系统工作在最佳状态。

软/硬件协同设计要求首先明确系统功能需求,从需求出发进行目标系统构思,经过构思完成系统的规范描述,然后在选定的开发环境中进行系统设计、仿真验证和综合实现等工作。

1.1 TCU功能分析

图1为AMT控制系统示意图。TCU的主要功能是采集各种传感器信号,依据算法进行分析判断,在执行机构的配合下,完成车辆的起步和换挡动作。相关的功能大体可分为人机界面相关、执行机构相关、CAN总线相关、安全处理相关4部分。

1.1.1 人机界面相关部分

TCU一方面要接收用户的各种指令,例如驾驶模式选择(自动/半自动,城市模式,等等)、操纵手柄、刹车等信息,同时要显示当前车辆的挡位、模式等信息,并对驾驶员的不正确操作或其他警示情况通过蜂鸣器提醒。

1.1.2 执行机构相关部分

TCU直接控制的执行机构有换挡、选挡和离合器3部分。TCU通过离合器位置、选挡位置、换挡位置传感器以及离合器速度传感器来感知执行机构位置或指令的执行情况,构成闭环控制系统。

1.1.3 CAN总线相关部分

TCU通过CAN总线共享发动机的相关信息,例如发动机转速、节气门开度等。另外,TCU通过协议请求,在换挡过程的特定时刻实现发动机转矩控

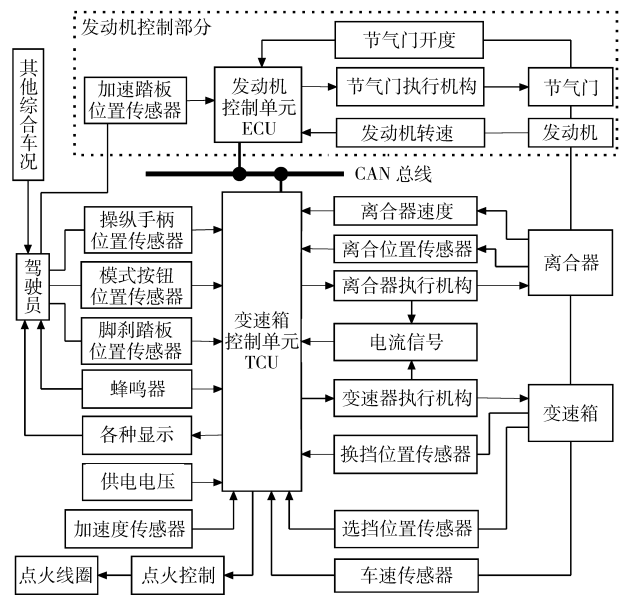


图1 AMT控制系统示意图

Fig.1 Control system of AMT

制,从而减小换挡冲击度,实现顺利换挡。

1.1.4 安全处理相关部分

安全问题包括:AMT本身的安全问题,例如要时刻监视执行电动机的工作电流,避免出现电动机烧毁情况;与车辆相关的安全问题,例如空挡之外不允许系统点火,从而规避潜在的危險。

1.2 协同设计平台及设计过程

要实现TCU的软/硬件协同设计,必须选择一个开发平台。Altera公司的开发平台Quartus II是一个全面的,易于使用且具有独立解决方案能力的软件,可以完成设计流程中从输入、综合、布局布线、仿真、时序分析和资源优化,以及编程、调试、验证的所有阶段。Quartus II中还集成了SOPC Builder工具,不需要直接编写HDL代码来定义系统,极大地节约了设计开发时间,开发方便。可以方便地在FPGA上嵌入一个或多个32位软核处理器Nios II系列,能从硬件结构上支持系统调试、程序下载、指令追踪和数据追踪,而不需要任何的BOOT程序。

如图2所示,在Quartus II软件环境中可以调用SOPC Builder,完成包括Nios II和各种IP模块的片上系统构建。生成系统后,可再次返回Quartus II系统中完成添加其他用户逻辑、编译、时序分析等工作,完成硬件构建过程;也可以进入Nios II IDE中开始设计C/C++应用程序代码。Altera提供外设驱动程序和硬件抽象层HAL(hardware abstraction layer),使用户能够快速编写与低级硬件细节无关的Nios II程序。即使在没有软件开发目标板的情况下,也可以经过编译、连接后通过Nios II指令仿真器运行和调试代码。

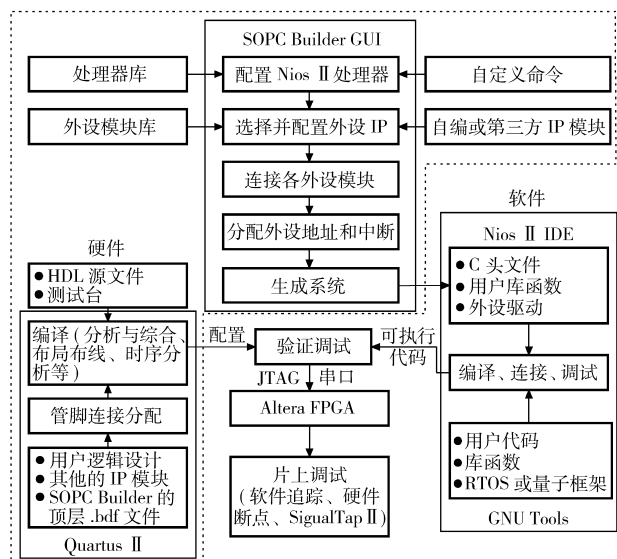


图 2 协同设计平台与设计过程框图

Fig. 2 Platform and processing of co-design

1.3 软/硬件划分及硬件设计

软/硬件划分是软/硬协同设计的关键问题之一,是指在设计系统时,确定各个模块是采取软件还是硬件的实现方式。软件的实现方案灵活,成本低;硬件的实现方案性能高,成本也较高。在一些包含复杂算法的系统中,可通过相关算法(如遗传算法、退火算法等)寻找软硬件的合理划分方案,以满足系统对时间、功耗、面积、成本等多方面的要求。

对 TCU 的软/硬件划分,除了上述提到的要素之外,必须考虑以下因素:①安全性和可靠性,汽车变速系统是驾驶安全息息相关的,硬件的可靠性相对较高。②前瞻性,汽车电子产品的开发和实验周期相对较长,随着半导体工艺的发展,FPGA 的价格和功耗在不断下降。③实用性,对于选定的 FPGA 芯片,要充分利用其剩余的容量,以减少软件的开销。鉴于上述因素,本设计选用性价比较高的汽车级产品 Cyclone II I 系列 FPGA,并将通常需要中断处理的速度计算模块,以及其他与硬件密切相关的底层接口模块用硬件来实现。主控制器从繁杂的底层运算和逻辑中解放出来,软件专注于策略判断与算法实现。

基于 SOPC 的 TCU 硬件设计分为两步:①与普通的嵌入式系统设计一样,实现 FPGA 与外部芯片和电路的正确连接,此时的 FPGA 仅仅是硬件系统的一个主要芯片。②构建 SOPC 系统,嵌入包括 Nios II 软核在内的 IP 模块,并通过 Avalon 总线连接,使 FPGA 在硬件逻辑上成为整个系统的控制核心。

构建 SOPC 的 IP 来自 3 个途径:EDA 厂商提供;用户自己定义;第三方提供^[5]。如图 3 所示,除

了 CAN_CTRL 模块(完成类似 SJA1000 协议控制器的功能)移植自 opencores 网站提供的开源 CAN 核^[6]外,其他的 IP 模块,要么直接利用 SOPC Builder 厂商提供的模块,要么用 VHDL 等硬件描述语言编写相应 IP 模块,并导入到 SOPC Builder 的用户 IP 库中进行调用。例如 SPEED_CAL、ADC_CTRL、DAC_CTRL、PWM_CTRL、SENSOR_CTRL 等模块都是用这种方式生成的。

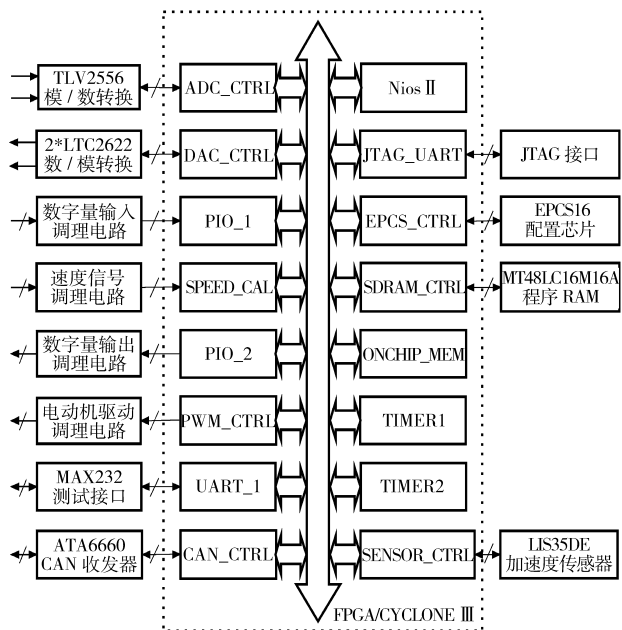


图 3 TCU 硬件示意图

Fig. 3 Hardware of TCU

SOPC 的运行需要扩展 FPGA 的外围支持电路,包括一些集成芯片和调理电路。芯片包括串行接口 A/D 转换器 TLV2556, 串行接口 D/A 转换器 LTC2622, CAN 收发器 ATA6660, RS232 电平转换器 MAX232 等。芯片 LIS35DE 为三维加速度传感器,以测量实车试验时的换挡冲击度。为了 SOPC 系统的正常运行,还需要外接一些必要的部件,如 JTAG 接口、串行配置芯片 EPCS16、程序运行用 RAM 等,这些部件在 FPGA 内都有相应的 IP 模块对应。另外,还需要添加运行时所需的其他外设,例如定时器、片上 RAM 等。

2 量子框架与总体结构

2.1 量子框架与软件总线

嵌入式系统一般都是事件驱动的。传统的前后台系统或 RTOS 系统都是顺序控制,使用循环和嵌套函数调用等标准结构,在程序的不同位置等待事件的发生,要么主动地挂起,要么被动地阻塞。尽管这种结构的编程方法在许多领域是可行的,但当存在多个可能事件,而且到达时间和顺序不定,又要求

实时处理时,其工作效果往往不是很好。一个比较好的解决方案就是将软件结构化为事件驱动编程模型。事件驱动系统中将应用分成若个事件,由事件监控框架等待事件的发生,然后将它们分发到句柄函数中进行事件处理。它们在运行过程中从一个状态转移到另一个状态,这些状态的变化和转移形成了状态机。

量子框架(quantum framework,简称 QF)是为嵌入式系统设计的轻量级、开源的状态机框架,其目的就是实时的方式来解决多异步事件的问题。不需要大的工具,QF 允许直接以 C/C++ 语言来编程实现并发层次状态机(UML 状态表)系统,由于 QF 完全控制了事件的生命周期和状态机,它可以提供传统 RTOS 所提供不了的功能,如零拷贝事件管理,出版-订阅事件分发,事件的自动垃圾收集,以及强大的软件追踪能力等。

AMT 系统是典型的嵌入式反应系统,其工作过程可以分为若干个状态,可以应用状态机的建模方法来建立 AMT 系统的行为逻辑模型。在 SOPC 的硬件架构上构建以量子框架为基础的事件驱动模型,将整个系统分为相互独立的活动对象。遵循下述两个规则:① 活动对象只通过异步事件来交互,而不是共享存储器或别的资源。② 活动对象在 RTC 处理过程中不应该被阻塞或处于“忙-等待”事件状态。显而易见,量子框架是作为“中间件”存在的,各活动对象之间没有直接联系,所有通信都通过量子框架来完成,量子框架在软件系统中扮演的是“软件总线”的角色。

2.2 总体结构

SOPC 的软/硬件结构与量子框架结合,形成了“五层/三总线/一框架”的软硬件综合总体结构,如图 4 所示。

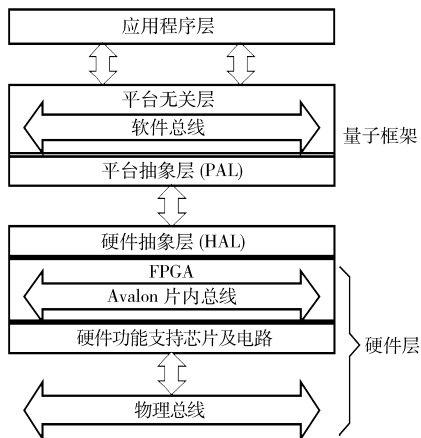


图4 五层/三总线/一框架结构图

Fig.4 Structure block of five layers/three buses/one framework

“五层”是指从下到上依次为硬件层(包括物理总线、硬件功能支持芯片和电路、FPGA)、硬件抽象层 HAL、平台抽象层 PAL (platform abstraction layer)、平台无关层和应用程序层;“三总线”是指物理总线(如 CAN/LIN/FlexRay/MOST/以太网等)、片内 Avalon 总线和软件总线;“一框架”是指量子框架。该总体结构具有如下特点:

(1) 层次分明,移植方便。在硬件和软件之间设置了硬件抽象层 HAL 和平台抽象层 PAL 两个缓冲层。HAL 屏蔽了硬件的细节,为 SOPC 系统的外围 IP 提供用户熟悉的 C/C++ 语言函数接口;PAL 屏蔽了量子平台运行硬件和软件环境的不同,一旦硬件平台发生改变,只要修改该层代码即可,PAL 层以上的所有层代码都不会受到影响。

(2) 互联和软硬件构建灵活。物理总线可以实现与其他电控模块互联,进行信息交流和控制协作;Avalon 总线可以连接多个 IP 模块,如果有 IP 需要加入或移除,不会影响其他已加入的 IP 模块;基于软件总线的用户程序模块,具有高内聚、低耦合的特性,添加和删除都比较方便。

(3) 事件驱动框架提高了开发效率,简化了开发过程,产品灵活性和鲁棒性更强。使用事件驱动框架的用户不必写主体程序,而是复用主体并写主体调用的代码。只要遵循相关规定,不需要在应用中使用互斥量、信号量等机制,QF 实时框架能够保证应用时不会有传统占先式多任务的缺陷,例如竞争、死锁、优先级反转、饥饿和非确定性等。

3 软件设计

对于基于量子框架的 AMT 开发步骤与开发内容,文献[2]已就 TMS320LF2407A 为核心的硬件平台和前后台(main + ISR)模式的有限状态机软件设计进行过较详细的论述。本文的研究对象为具有 5 个前进挡和 1 个倒挡的普通轿车,其开发步骤和开发内容与文献[2]中的重型卡车基本类似,此处不再赘述。本文与文献[2]有两点明显不同:使用了 Nios II 软核 CPU,涉及平台抽象层 PAL 的移植问题;设计了主活动对象的层次状态机 HSM,比传统的有限状态机 FSM 更加适合于 AMT 系统。

3.1 基于 Vanilla 内核及 Nios II 的移植

量子平台内嵌两个内核,一个为“Vanilla”非占先式内核,另一个为 QK 占先式内核。非占先式内核的任务级响应等于系统中的所有活动对象中最长的 RTC(run-to-completion)步骤。由于不需要等待事件,Vanilla 内核的任务级响应好于传统的长循环(main + ISR)。由于该内核简单而有效,因此

AMT 的软件设计中使用了 Vanilla 内核。

由于 QF 包含一个定义清晰的平台抽象层, 包装了所有平台相关代码, 使之与平台中心代码相分离, 所以移植 QF 相对容易。与移植相关的头文件有 2 个, 即 qep_port.h 和 qf_port.h 头文件。此外, 还需要对中断服务程序进行配置。

3.1.1 qep_port.h 头文件

该头文件用来修改和配置 QEP 事件处理器, 是将 Altera 的 HAL 提供的数据类型表示方式与 WG14/N843 C99 标准对应, 文件主要内容如下:

```
#include "alt_types.h" /* Altera 硬件抽象层的类型定义 */
```

```
typedef alt_8      int8_t;
typedef alt_16     int16_t;
typedef alt_32     int32_t;
typedef alt_u8     uint8_t;
typedef alt_u16    uint16_t;
typedef alt_u32    uint32_t;
```

```
#include "qep.h" /* QEP 平台无关公共接口 */
```

3.1.2 qf_port.h 头文件

该头文件中定义了活动对象的最多数目、临界区的中断锁定和解锁策略、其他头文件, 其主要内容如下:

```
#define QF_MAX_ACTIVE 63 /* 活动对象的最多数目 */
```

```
#define QF_INT_KEY_TYPE alt_irq_context
#define QF_INT_LOCK(key_) ((key_) = alt_irq_disable_all())
```

```
#define QF_INT_UNLOCK(key_) alt_irq_enable_all(key_)
```

```
#include "sys/alt_irq.h" /* for alt_irq_disable_all()
                          /alt_irq_enable_all()
                          */
```

```
#include "qep_port.h" /* QEP 移植 */
#include "qvanilla.h" /* "Vanilla" 内核 */
#include "qf.h" /* QF 平台无关接口 */
```

3.1.3 中断服务程序

Altera HAL 提供了低级中断包装函数 alt_irq_handler()。Vanilla 量子平台移植可以使用标准 HAL 的中断实现, 但使用前需要 alt_irq_register() 函数来登记要处理的每一个中断。

QP 移植中需要提供一个时间源来周期性地调用 QF_tick() 函数, 以处理准备好的时间事件。而调用 QF_tick() 最理想的地方就是时间周期中断。由于时间周期中断已经被 HAL 登记和使用, 所以要做的最主要工作就是在时间周期中断中添加一个“钩

子”即可。

在 alt_hooks.h 头文件中添加:

```
#define ALT_OS_TIME_TICK() QF_tick()
```

并在该文件中增加一个函数声明:

```
void QF_tick(void)
```

3.2 主活动对象的层次状态机设计

量子框架兼容有限状态机 FSM 和层次状态机 HSM。对于比较复杂的系统, 如 AMT 系统, 使用 FSM 设计会带来“状态爆炸”的问题, 使用 HSM 则较好地解决了这一问题。

根据车辆的具体运行情况, 定义两个活动对象 QActiveShift 和 QActiveMain^[2], QActiveMain 为主活动对象, 其层次状态机如图 5 所示。根据发动机是否启动, 可以得出 2 个总状态, 即发动机熄火状态 engine_off 和发动机运转状态 engine_on。这 2 个状态又包括了若干子状态, 其中 engine_off 状态又分为 forward_gears、reverse_gear 和 neutral 子状态; engine_on 状态又分为 car_stop、car_startup 和 car_run 子状态。每一个子状态又可以分为若干子状态。

状态机转换解释如下:

(1) v_1 为发动机运转速度的下限值, δ 为发动机回差值, 当 engine_speed $> (v_1 + \delta)$ 时, 就意味着发动机点火成功。相反, 当 engine_speed $< (v_1 - \delta)$ 时, 就意味着发动机已经熄火。

(2) 在 engine_off 状态, 可以对 AMT 进行换挡测试。用户的操纵手柄会产生前进挡事件 UP_SHIFT、倒挡事件 DOWN_SHIFT 和空挡事件 N_SHIFT, 从而引发升挡行为 up_shift()、倒挡行为 down_shift() 和换空挡行为 n_shift()。要注意, 只有空挡才允许打火操作 (ENABLE_ignition), 在前进挡和倒挡都不允许打火操作 (DISABLE_ignition)。

(3) 发动机运转状态的初始路径是进入静止状态 car_stop, 进入静止状态后的初始路径是空挡状态 neutral。在 car_stop 内允许前进挡 gear_1、倒挡 reverse_gear 及空挡 neutral 之间的相互转换。当在 reverse_gear 和 gear_1 时, 如果加速踏板被踩下, 则进入起步状态 car_startup, 如果车速为零且加速踏板深度为零, 则从起步状态返回静止状态 car_stop。

(4) 汽车起步状态 car_startup 又分为 3 个子状态: clutch_early、clutch_mid 和 clutch_late。其中, clutch_early 指从离合完全分离到离合初次接触阶段; clutch_mid 是指离合器滑摩阶段; clutch_late 指离合结束滑摩到完全结合阶段。其转换条件是由发动机转速、油门踏板深度和速度 (加速度)、车速、刹车踏板等多个变量构成。起步过程是可逆的, 不同的标准 (cond_1, cond_2, cond_3, cond_4) 对应不同

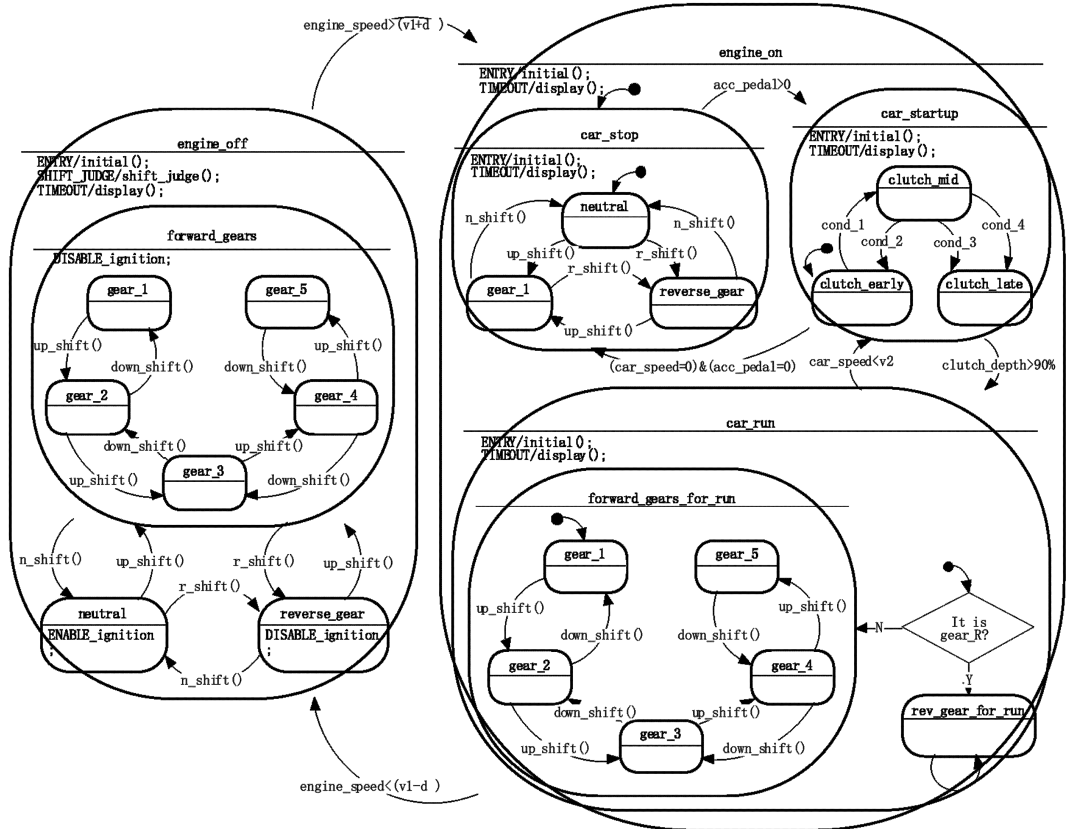


图5 主活动对象层次状态图

Fig.5 HSM of main active object

的起步算法,也往往决定了不同的起步质量。

(5) 汽车起步成功,会转到汽车运行子状态 car_run。如果在倒挡起步,则一直都在倒挡中运行,直到退出 car_run 状态为止;如果在前进挡起步,则会在 1 挡~5 挡之间相互转换,直到退出 car_run 状态为止。

由此可以看出,层次状态机使整个状态转换结构更加清晰,例如汽车在起步状态 car_startup 中熄火,由于行为继承,则可以跳转到发动机熄火状态 engine_off 重新开始。

4 实车测试

为了验证整体系统,特别是在量子框架下的主活动对象层次状态机的有效性,对图 6 所示的 AMT 系统进行了实车测试。

使用 USB7325 采集板卡采集 AMT 工作中的各种参量,使用 USB/CAN 转换器采集 CAN 总线数据,并通过便携式计算机随车记录数据,监控程序用

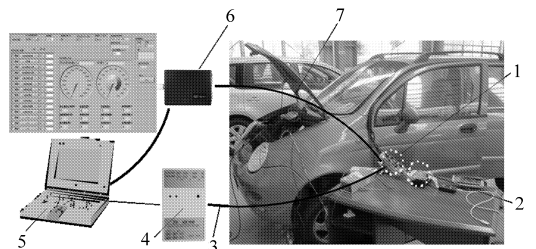
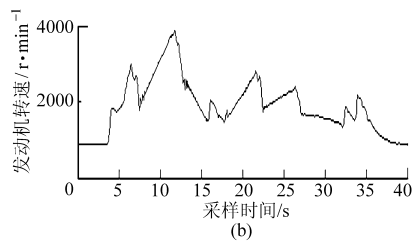
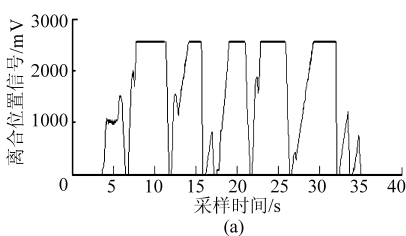


图6 数据采集实验平台

Fig.6 Platform of data collection

- 1. 分线端 2. TCU 3. 模拟及数字采集电缆 4. USB7325 数据采集卡 5. 便携式计算机 6. USB/CAN 转换器 7. CAN 总线

LabVIEW 编写。对停车、点火、起步及行驶过程的各种状态进行实验表明:主活动对象的层次状态机工作正确有效。图 7 所示为汽车从 1 挡起步,经过 2 次加速,最高升到 4 挡,又从 4 挡下降到 1 挡的全过程记录。试验是在路况较好的平整沥青路面上进行的。每次换挡都会伴随离合器的动作;挡位状态变化与油门踏板深度、发动机转速等参量有关,与设计意图相符合。



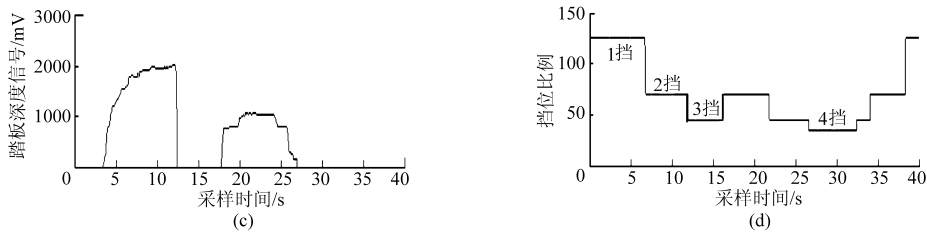


图 7 前进挡起步及换挡实例

Fig. 7 Example of forward startup and shift

5 结论

(1) 将软/硬件协同设计的思路引入到 AMT 控制系统的设计中,提高了电控系统开发的速度,增加了开发灵活性。

(2) 将量子框架的软件总线概念引入到软件设

计框架中,提高了软件的质量,增加了软件的开放性、复用性和可移植性。

(3) 针对复杂系统的层次状态机设计,充分利用行为继承优势,避免有限状态机的“状态爆炸”问题,使整个控制结构思路清晰,层次分明。

参 考 文 献

- 冯能莲, 李克强, 连小珉, 等. 液力机械传动车辆自动换挡控制系统[J]. 农业机械学报, 2004, 35(1): 8~12.
Feng Nenglian, Li Keqiang, Lian Xiaomin, et al. Study on automatic-shift control system of a vehicle equipped with hydro-mechanical transmission[J]. Transactions of the Chinese Society for Agricultural Machinery, 2004, 35(1): 8~12. (in Chinese)
- 李晓军, 张承瑞. 重型卡车自动变速器的控制器设计[J]. 农业机械学报, 2008, 39(8): 18~23.
Li Xiaojun, Zhang Chengrui. Design of heavy truck AMT controller based on quantum framework[J]. Transactions of the Chinese Society for Agricultural Machinery, 2008, 39(8): 18~23. (in Chinese)
- 李洪斌, 张承瑞. 基于量子框架的开放式汽车电控系统体系结构[J]. 吉林大学学报, 2006, 36(2): 166~171.
Li Hongbin, Zhang Chengrui. Open architecture of automotive E/E control system based on quantum frame[J]. Journal of Jilin University, 2006, 36(2): 166~171. (in Chinese)
- Qi Shengbo, Zhang Chengrui, Qi Zhenjun, et al. SOPC technology applied on automatic mechanical transmission control unit design[C]//Proceedings of the 8th World Congress on Intelligent Control and Automation, Ji'nan, China, 2010.
- 方茁, 陈泽文, 彭澄廉. SOPC 设计中的用户自定义逻辑[J]. 计算机工程, 2004, 30(17): 42~44.
Fang Zhuo, Chen Zewen, Peng Chenglian. User-defined logic in SOPC design[J]. Computer Engineering, 2004, 30(17): 42~44. (in Chinese)
- 戚振军, 张承瑞, 綦声波. 基于 AFS600 的 CAN 核重用设计[J]. 机械与电子, 2010(5): 41~44.
Qi Zhenjun, Zhang Chengrui, Qi Shengbo. Design of reusing CAN IP core on AFS600 [J]. Machinery & Electronics, 2010(5): 41~44. (in Chinese)
- 葛保建. 基于 SOPC 的软硬件设计平台的研究与实现[D]. 武汉: 武汉科技大学, 2008.
Ge Baojian. The research and implement of hardware/software co-design platform based on SOPC [D]. Wuhan: Wuhan University of Science and Technology, 2008. (in Chinese)
- 徐海涛. 基于 SOPC 的软硬件划分算法研究[D]. 哈尔滨: 哈尔滨理工大学, 2009.
Xu Haitao. Research on hardware/software partitioning algorithm based on SOPC [D]. Harbin: Harbin University of Science and Technology, 2009. (in Chinese)
- Keating M, Bricaud P. Reuse methodology manual for system-on-a-chip designs[M]. 3rd ed. Boston: Kluwer Academic Publishers, 2002.
- Miro Samek. Practical UML Statecharts in C/C++: event-driven programming for embedded systems[M]. 2nd ed. Newnes, 2008.
- 李晓军, 张承瑞, 杨泉. 软件总线在 AMT 电控单元中的应用研究[J]. 汽车工程, 2009, 31(4): 304~307.
Li Xiaojun, Zhang Chengrui, Yang Quan. A study on the application of software bus to AMT ECU [J]. Automotive Engineering, 2009, 31(4): 304~307. (in Chinese)
- 张俊智, 薛俊亮, 潘凯. 混合动力系统控制软件的开发[J]. 机械工程学报, 2009, 45(5): 115~120.
Zhang Junzhi, Xue Junliang, Pan Kai. Development of hybrid powertrain control software[J]. Chinese Journal of Mechanical Engineering, 2009, 45(5): 115~120. (in Chinese)