CHIP OS: NEW ARCHITECTURE FOR NEXT GENERATION EMBEDDED SYSTEM

Tianzhou Chen, Yi Lian, Wei Hu
College of Computer Science, Zhejiang University
Hangzhou, Zhejiang, 310027
P. R. China
{tzchen, yl, ehu}@zju.edu.cn

ABSTRACT
Nowadays embedded system, hardware/software technology has progressed prosperously. In many field of industrial manufacture and people life, embedded system is indispensable. Recently hardware of embedded system becomes more powerful and sophisticated. SOC is an important trend for hardware design. New software architecture has to be proposed to adapt to this progress. In this paper, new software architecture for embedded system is proposed and described.

KEY WORDS
Embedded system, architecture, power-efficient, real-time, chip, SOC

1. Introduction

Embedded system has provided increasing sophisticated functions in recent years. The embedded system will have rich peripherals to give user novel services. GPS, Cameras, Bluetooth, USB, Wireless LAN, SD and IC cards, are been used as extended devices. In addition to the rich peripherals, software functions in the embedded system increase rapidly. The followings will be the key components of current and future software functionalities of embedded system. DBMS, JAVA, GUI, Communication, Internet Access. And this software platform could be used in the follow fields: digital household appliance, mobile terminal, automobile electronics and digital instrument.

The traditional software architecture of embedded system is similar to normal computer. Software platform consists of normal operation system, middleware and user application. Normal operating system manages hardware resource and serves for bridge between user and hardware. Depending on application function, middleware consists of DBMS, Java, Internet access and so on. Some previous solution may integrate normal operating system and middleware into embedded software platform.

Recently hardware of embedded system becomes more powerful, especially chip. Chip manufacturer has integrated much hardware into single chip, such as SRAM, Flash and WLAN. By the help of the progress of chip manufacture, more and more hardware is intended to be integrated into chip, as shown in Figure 1. A sophisticated software architecture is needed to adapt to this progress and take advantage of chip manufacture. In this paper, the requirement of new architecture will be summarized and Chip OS architecture is described.

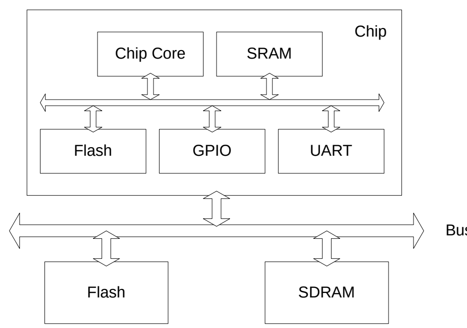

Figure 1: new architecture for embedded hardware

2. Requirement for Chip OS

Current embedded OS is often an integrated software platform, including all function together. They store in flash or hard disk, loaded into RAM on board when system start. But it has some disadvantages when it is used for embedded system. The Chip OS contraposes these disadvantage to make optimization. The following is the characteristics of Chip OS:

RAM usage: Usually OS kernel boots from ROM area or file system in disk, and its memory images are expended in RAM area of system and executed instructions in it. RAM is one of cost factor in embedded system and reducing the size of RAM is always required. Chip OS is modularized. The kernel is very small and the module will be inserted into kernel dynamically dispending on application requirement.

Power efficiency: A large number of embedded systems are driven by batteries, not by wired power supply. Efficient power usage is required to increase utilization time. Normal embedded OS makes use of DPM or other optimized algorithm to reduce power consumption. But previous platform architecture must use SDRAM on board as memory. And SDRAM costs power very much. Chip OS store in flash and run in SRAM on chip. The ratio of power consumption for SRAM is expected to be 10% of SDRAM. So power efficiency of new architecture will be more excellent than previous architecture.

Real-time Response: Many embedded applications require real-time processing. Current OS architecture use software method to guarantee real-time processing. The technology of process preemption, process priority is often used in software solution for real-time computing. Chip OS provide OS API to application developer. But the implementation of Chip OS is hidden. Some part of Chip OS may be implemented by hardware, for example, task scheduler may be implemented by FPGA. The real-time response time will be descending acutely.

3. Architecture of Chip OS

Chip OS proposes a new architecture for embedded system with SOC. This architecture is summarized in Figure 2. It is a module-based architecture. And the fundamental principle for Chip OS is: Make sure the kernel run in SRAM minimum; Make sure the scheduler of modules is optimal. This principle would be further described below. Chip OS architecture consists of two main parts, Microkernel on chip and Module Management on flash:

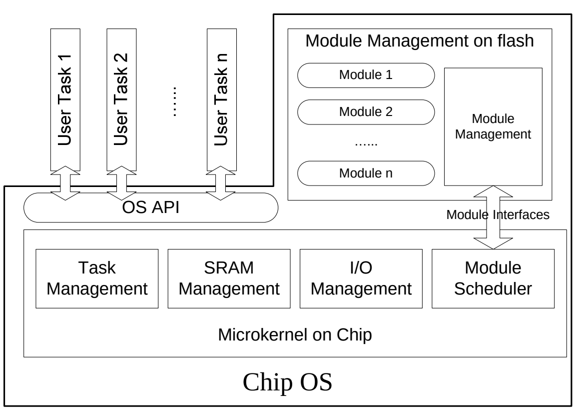

Figure 2: Architecture of Chip OS

Microkernel on Chip

This part stores in flash on chip when system shuts down. When start up, microkernel is loaded into SRAM on chip. For the size of SRAM is limited relatively, microkernel is required to be minimum. Four basic components constitute microkernel. They are further described below:

Task Management takes charge of tasks through OS API from users. These tasks are scheduled depending schedule criterions, such as highest CPU utilization or real time. The implementation of this part is hidden from users. The alternatives may be hardware, software or cooperative hardware/software kernel. For example, if hard real time is required, this task scheduling could be implemented by hardware or FPGA so that the switch of task could be fast.

SRAM Management: microkernel runs in SRAM on chip rather than SDRAM on board. This solution is power-efficient oriented. SRAM is expected to 10% of energy consumption as SDRAM. But the SRAM is more expensive, so the size of SRAM may be relatively small than SDRAM on board. Modularization and module scheduling dynamically make kernel small enough to be loaded and run in SRAM. More, SRAM on Chip is constituted by several banks, e.g. four banks, SRAM management may turn on/off some of them to reduce power consumption.

I/O Management: The embedded system will have rich peripherals to give user novel services. GPS, Cameras, Bluetooth, USB, Wireless LAN, SD and IC cards, are been used as extended devices. Chip OS manage these I/O operation of device on chip. This part has few differences from normal embedded operation systems.

Module Scheduler: one of principles of Chip OS design is minimum microkernel on chip. It is guaranteed by Module scheduler. Module scheduler switches the module useless out, switches the required module in and insert it into kernel. This process is dynamic. In real time system, modules should be grouped into real-time relative and real-time irrelative. Real-time relative modules won’t be switch out.

Modules Management on Flash

All modules are stored in flash on Chip or removable devices. And modules are indexed in B+ tree, so that module could be queried and accessed quickly. The management module is called by module scheduler in microkernel. Module scheduler asks for the necessary modules and inserts them into microkernel. More, modules management could be in removable devices, so that users could change GUI, DBMS or other tools through changing removable cards, e.g. SD cards. This figure makes mobile terminal more portable.

Chip OS API:

Chip OS provide a uniform API to user application developers. Different chip manufacturer should follow this uniform API. But they could change and improve the implementation in chip. User tasks call these APIs and collaborate with Chip OS.

4. Conclusion and future work

In this paper, new architecture, Chip OS, is introduced for embedded system. The Chip OS is designed for embedded system. It has much technology that attracts the embedded system. It provides efficient RAM usage, low power consumption and potential for good real-time response. The design makes embedded system more mini, more portable. In future work, some detail implementation in Chip OS may be further improved and the architecture will fit the requirement of embedded system well more.

References

[1] Sara Berg, Alex S. Taylor, Richard Harper "Mobile phones for the next generation: device designs for teenagers" Human Factors in Computing Systems 2003

[2] J. Lapalme, E. M. Aboulhamid "a new solution for embedded systems modeling and simulation" Proceedings of the 2004 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools 2004.

[3] Fabrice Mérillion Gilles Muller "Dealing with Hardware in Embedded Software: A General Framework Based on the Devil Language" Proceedings of the ACM SIGPLAN workshop on Languages, compilers and tools for embedded systems table of contents 2001

[4] Dexin Li, Qiang Xie and Pai H. Chou Center for Embedded Computer Systems University of California, Irvine: Scalable Modeling and Optimization of Mode Transitions Based on Decoupled Power Management Architecture 2003

[5] Camil Demetrescu, Irene Finocchi "Software engineering: applications, practices tools (SE): A portable virtual machine for program debugging and directing" March 2004 Proceedings of the 2004 ACM symposium on Applied computing

[6] Qing-Li Zhang, Ming-Yuan Zhu, Shuo-Ying Chen "Technical correspondence: Automatic generation of device drivers" June 2003