A Study of Fast Context Switching between Register-Sets, at Embeded RTOS

Tadashi Ogino (110027)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 13, 2003

Keywords: Embeded, Real Time, Real Time OS, Context Switch.

Abstract

Nowadays, Real Time OS is applied at systems which need fast processing, but at giga-bit communication system, the over-head which is context switching from ten μ to μ seconds wasted, causes serious problems. It is made from cache miss by memory accessed when task context moving. We propose two approaches, one is *Context switching between register-sets*, another is *Narrow down registers user task using*, on CPU with several register-sets, evaluate by simulation.

1 Background and purpose

Now, the overhead of context switching in RTOS, from ten μs to μs seconds needed. It is allowed at mili-seconds deadline user task, at μ seconds deadline task, makes tasks deadline-overed possiblity. (example : The real-time system at giga-bit communication system, make one packet proceccing in μ seconds. So μ seconds overhead makes possible deadline overed, a packet is losted.)

We focus memory accessing at context switching made overheads, for fast context switching, propose two methods below and evaluate them.

- Context switching between register-sets
- Narrow down registers user task using

2 Context switching between register-sets

To reduce data-cache miss at context sitching, context swithing should be terminated in CPU. So we choose Casablanca[1], with 8 register-sets and data moving instruction between register-sets, register-set number 2 to 6 are avairable *Context saved register-set*, make it realized.

It makes improved followings:

1. Fasten user task context switching
2. Goodness of data chache usage

3 Narrow down registers user task using

Formerly we have moved whole a register-set by context switching. It is insured task context against context swithcing, but needless unused register data moving. So we make it realized by added an infomation *Register usages* in user task.

We change context saved register-sets with it. We manage them as frames composed 16 registers. Each frames are managed by "frame management tables" on main memory, it is writen *task-id* when frame is used, else witten *0*.

It makes improved followings:

1. Much contexts available in context register-sets
2. Reduce instructons at context swithcing
3. More goodness of data chache usage

4 Conclusion

We evaluated proposed and historical methods on simulation, improved followings:

1. Reduce memory accessing at context swithcing
2. Reduce data-cache missing as it

3. Reduce deadline-overed tasks

4. Shoten user task response

References

[1] Kiyofumi Tanaka, Takashi Matsumoto: "Casablanca: A Real-Time RISC Core for Embedded Systems", International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, 2001, CD-ROM.