Embedded Systems Clock Optimization for Low Power

Zdravko Karakehayov
Technical University of Sofia
zgk@computer.org

Ludmila Karakehayova
Technical University of Sofia
lvk@vmei.acad.bg

ABSTRACT

This paper addresses the design of control-dominated embedded systems for low power. The power consumption can be minimized at technological, architectural and system level. Using microcontrollers for embedded systems design we can employ power management to improve the situation at system level. We present a method for system clock optimization based on a power saving mode and behavioural requirements. Most obviously, the suggested approach minimizes the power consumption. A more subtle effect is improvement of the electromagnetic compatibility (EMC). In parallel with the longer battery lifetimes, the designs are capable of achieving higher levels of integration.

Keywords *- Embedded systems, microcontrollers, power minimization, scheduling.*

1. Introduction

A taxonomy of methods for low power design would include three major branches: IC technology, architecture and system level approaches. As far as the architecture is concerned, we distinguish between synchronous and asynchronous circuits. A powerful method for low power optimization of synchronous circuits is discussed in [Alid 94]. The method is based on selectively precomputing the output values of the circuit one clock cycle before they are required and consequently reducing the internal switching activity in the following clock cycle. Alternatively, the power dissipation can be declined by reduction of the average number of bit changes per state transition [Hach 94].

Using microcontrollers for embedded systems the designers can benefit from the power management modes. Also, the power consumption is closely related to the microcontroller oscillator frequency.

2. Calculating the minimal oscillator frequency

There are two factors which define the floor of the microcontroller's clock rate. First, the oscillator frequency must exceed a certain minimum which is specified by the manufacturers for dynamic devices. Second, the use of a serial port dictates a series of specific values. With the assumption that 8051 microcontrollers are concerned, the following relationship emerges

$$f_{OSC} = 12(256 - TH1)(32 - 16(SMOD))BR$$

f_{OSC} (MHz)	256 - TH1				
	1	2	3	4	5
1200 bps	0.2304	0.4608	0.6912	0.9216	1.1520
2400 bps	0.4608	0.9216	1.3824	1.8432	2.3040
4800 bps	0.9216	1.8432	2.7648	3.6864	4.6080
9600 bps	1.8432	3.6864	5.5296	7.3728	9.2160

Figure 1 The oscillator frequency as a function of the baud rate and the auto-reload value TH1

This equation is relevant for serial port mode 1 or 3 and Timer 1 mode 2. The oscillator frequency depends on the code in the register TH1, the bit SMOD and the baud rate (BR). Figure 1 shows a few calculations for the clock rate. Furthermore, Figure 2 represents graphically the data from Figure 1. It shows how different baud rates can be achieved combining oscillator frequencies and auto-reload codes.

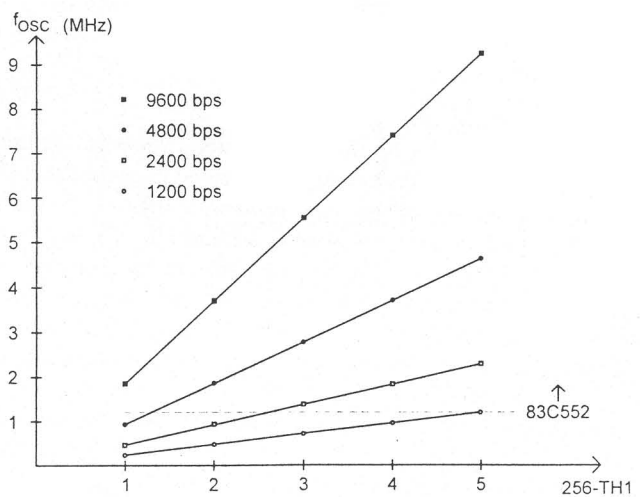

Figure 2 Oscillator frequencies for different baud rates

3. Using behavioural requirements for clock optimization

We assume that a control-dominated embedded system runs in a cyclical manner as shown in Figure 3.

The specification requires the microcontroller to execute instructions for a certain number of clock cycles (NC). This is the active period (T_{ACT}). As soon as the tasks have been completed, the microcontroller can enter a power saving mode and

stay there till the end of the cycle. Note that the microcontroller communicates over a serial link for a period of time (T_{COMM}) which can overlap both the active and idle phases. In fact, the communication requires interrupts on the background of the power saving mode, but the microcontroller is activated for relatively short periods of time and they can be neglected.

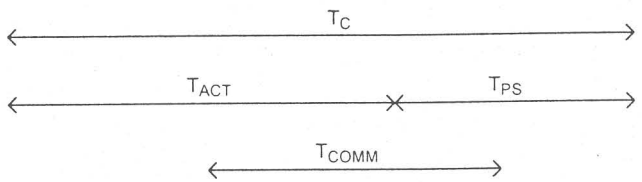

Figure 3 The embedded system cyclical pattern

The energy consumed by the microcontroller is given by

$$E = PT = IV_{CC}T = \{ I = kf_{OSC} + n \} =$$

$$= (k_{ACT}f_{OSC} + n_{ACT})V_{CC}\frac{NC}{f_{OSC}} + (k_{PS}f_{OSC} + n_{PS})V_{CC}\left(T_C - \frac{NC}{f_{OSC}}\right) =$$

$$= k_{PS}V_{CC}T_Cf_{OSC} + (n_{ACT} - n_{PS})V_{CC}NC\frac{1}{f_{OSC}} + (k_{ACT} - k_{PS})V_{CC}NC + n_{PS}V_{CC}T_C \quad (1)$$

The energy has a minimum for

$$f_{OSCOPT} = \sqrt{\frac{(n_{ACT} - n_{PS})NC}{k_{PS}T_C}} \quad (2)$$

Different microcontrollers will have different relationships between the clock rate and the supply current. For example, Figure 4 shows power supply currents in active and Idle modes for three microcontrollers produced by Philips. The 80CL51 microcontroller has been especially designed for low power applications. The hallmark of the 83C552 microcontroller is a set of sophisticated embedded peripherals such as an ADC, pulse width modulated outputs and I^2C interface. The 51XAG1 microcontroller is a 16-bit machine with two enhanced UARTs.

80CL51 Philips	Active	$I = 0.88 \times 10^{-9} f_{OSC} + 1.2 \times 10^{-3}$
	Idle	$I = 0.26 \times 10^{-9} f_{OSC} + 0.5 \times 10^{-3}$
83C552 Philips	Active	$I = 2 \times 10^{-9} f_{OSC} + 3 \times 10^{-3}$
	Idle	$I = 0.44 \times 10^{-9} f_{OSC} + 0.4 \times 10^{-3}$
51XAG1 Philips	Active	$I = 2.2 \times 10^{-9} f_{OSC} + 25 \times 10^{-3}$
	Idle	$I = 0.83 \times 10^{-9} f_{OSC} + 0.5 \times 10^{-3}$

Figure 4 Power supply currents in active and Idle modes

The clock optimization method goes through the following steps;

- Find the optimum oscillator frequency f_{OSCOPT}.
- If the following condition is met

$$\frac{NC}{f_{OSCOPT}} > T_C \tag{3}$$

then increase the oscillator frequency to accommodate the required number of clock cycles within the specified control cycle T_C.

- Increase the oscillator frequency eventually to achieve a standard value which is consistent with the required baud rate (BR).

For instance, the specification of an 83C552 based embedded system demands a control cycle of 2 ms, 1000 clock cycles to process the tasks and communication capability consistent with baud rate 4800 bps.

The optimal oscillator frequency

$f_{OSCOPT} = 1.719$ MHz

The next step is to check if the execution time is not longer than the control cycle.

$$\frac{NC}{f_{OSCOPT}} = 581.7\ \mu s < 2\ ms$$

Finally, the specified baud rate of 4800 bps requires the oscillator frequency to be increased to 1.8432 MHz. The corresponding auto reload code is 254.

4. Conclusion

Small-scale, control-dominated embedded systems typically operate in a cyclical manner. As is frequently the case, the system specification includes the control cycle duration, a work load measured in clock cycles and a specific baud rate. The suggested approach brings down the power consumption by optimizing the oscillator frequency. The optimization procedure begins with calculation of the clock rate which leads to a minimal power consumption and completes with a solution consistent with the specified timing parameters.

5. References

[Alid 94] Mazhar Alidina, Jose Monteiro, Srinivas Devadas, Abhijit Ghosh and Marios Papaefthymiou, "Precomputation-based sequential logic optimization for low power", *Digest of Technical Papers*, pp. 70-73, IEEE/ACM International Conference on Computer-Aided Design, November, 1994.

[Hach 94] Gary D. Hachtel, Mariano Hermida, Abelardo Pardo, Massimo Poncino and Fabio Somenzi, "Re-encoding sequential circuits to reduce power dissipation", *Digest of Technical Papers*, pp. 70-73, IEEE/ACM International Conference on Computer-Aided Design, November, 1994.

[Laks 99] Ganesh Lakshminarayana and Niraj K. Jha, "High-level synthesis of power-optimized and area-optimized circuits from hierarchical data-flow intensive behaviors", *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, No. 3, March 1999, pp. 265-281.

[Niel 99] Lars S. Nielsen and Jens Sparso, "Designing asynchronous circuits for low power: an IFIR filter bank for a digital hearing aid", *Proceedings of the IEEE*, vol. 87, No. 2, February 1999.

[Phil 96] Philips Semiconductors, *16-Bit 80C51XA Microcontrollers, Data Handbook IC25*, 1996.

[Phil 97] Philips Semiconductors, *80C51-Based 8-Bit Microcontrollers, Data Handbook IC20*, 1997.