Embedded Systems Design with Microcontrollers and FPGA

Zdravko Georgiev Karakehayov
Technical University of Sofia
zgk@computer.org

ABSTRACT

Embedded systems design with microcotrollers and FPGA *begins with outlining the desired functionality. The system specification is usually combined with a particular model. A Data Flow Graph (DFG) is a suitable model for data-dominated systems. Control Flow Graphs (CFG) have often been used with control-dominated systems. Also, Control-Data Flow Graphs (CDFG) can be used as intermediate representation of a behavioural description. The next phase is to find an architecture which is most suitable for the specific application. In case of a single processor architecture the system can be enhanced by an ASIC. The ASIC is used to improve the timing parameters of the embedded system. While the processor core typically belongs to a microcontroller, the ASIC implementation is associated with FPGAs.*

The paper addresses the implementation of the communication buffers and trade-offs which come with synchronisation procedures.

Keywords *- Embedded systems, microcontrollers, FPGA.*

1. Introduction

The embedded systems design process starts with specifying the desired functionality. Specification languages such as VHDL, Verilog and SDL are widespread and suitable for a large spectrum of applications [Arms 1993, Gajs 1994]. At this phase, the system specification is combined with a particular model. Figure 1 shows the initial design phases [Kara 1999a]. The indicated models are Finite State Machine (FSM), Data Flow Graph (DFG) and Control Flow Graph (CFG). The next phase is to find an architecture which is the most suitable for the specific application. After the system specification, the design process goes on with partitioning, allocation and mapping.

There are two approaches to system partitioning. Structural partitioning implies that the system structure is defined first and then partitioned. Structure is an interconnection of hardware objects (modules). This is a straightforward method for evaluation of the size and pin number. Alternatively, the functional partitioning decompose the system's functionality into functional objects. The imperative advantage of the functional partitioning becomes visible when the hardware/software trade-off is approached. Since the objects (tasks) are functional, they can be implemented either in hardware or software.

The mapping is the phase where communication links emerge. As a result of the allocation and mapping, communication between microcontrollers and ASIC has to be organised. Performance requirements demand parallel interface to link both types of devices.

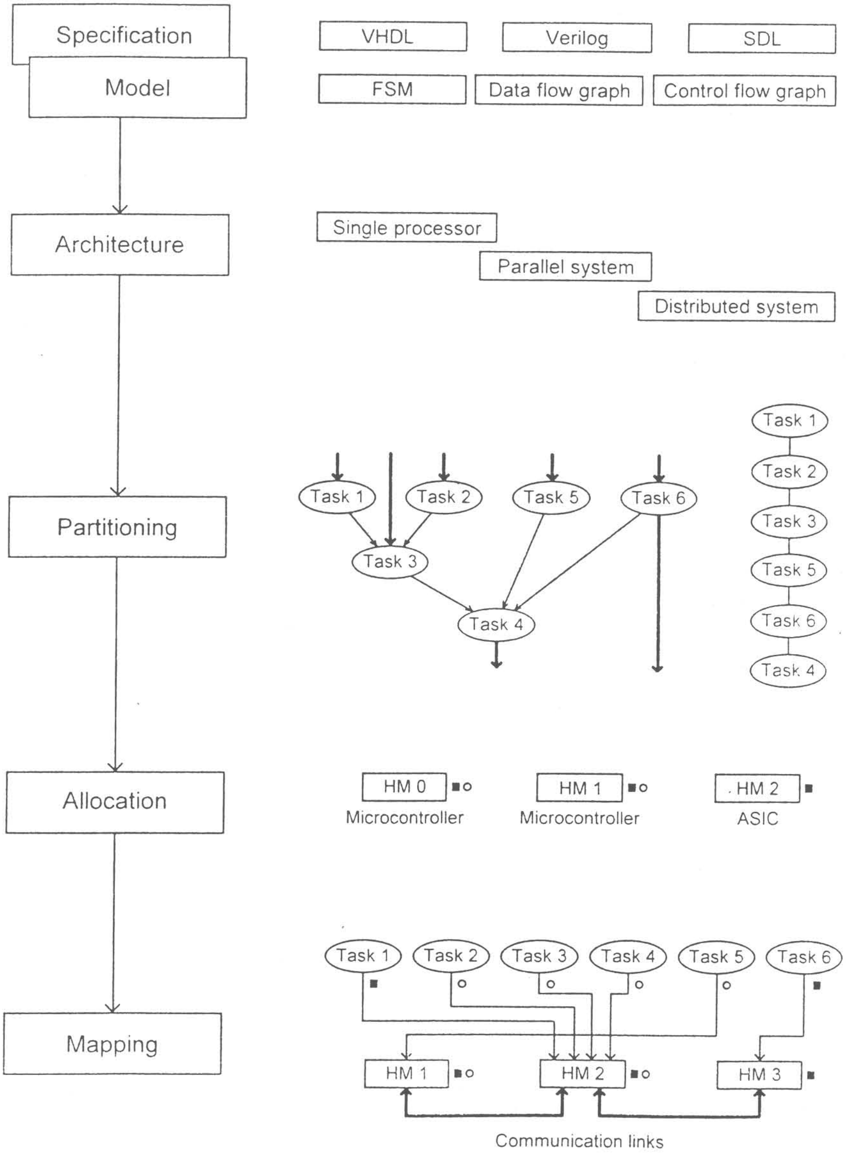

Figure 1 Design phases

2. ASIC architecture

The ASIC architecture usually vary between a strait forward implementation of FSM and datapath-controller designs. The processor's datapath frequently includes ALU, multipliers, shifters and registers. However, the control-dominated embedded applications may require different types of processing elements. The EPROM emulator example, shown in Figure 4, includes three multiplexers in its data path.

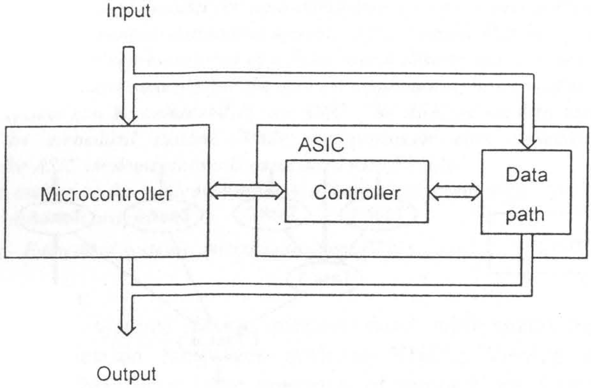

Figure 2 The embedded system architecture

3. Communication between ASIC and microcontrollers

There are applications when ASIC is used not only as an accelerator, but also as a communication channel which links microcontrollers. We can face this situation if the hardware modules HM2 and HM3 are exchanged (Figure 1).

The communication requires buffers and the size of the buffers is crucial. Furthermore, different approaches can be used to interact with the buffers. Full buffers and empty buffers can initialise read and write operations. Also, trigger levels can be programmed to improve the efficiency. The trigger level approach should be combined with time-out timers. Time-out timers help the system to avoid deadlocks.

Figure 3 shows the transition from a rudimentary syncronization procedure to a sophisticated interaction. Introducing a trigger level, which activates the communication is the first improvement. This approach promises to minimize the probability of overrun errors. However, it also creates a new threat. If the piece of information exchanged is below the trigger level, the interaction will be blocked. This problem is overcome by introducing a time-out period.

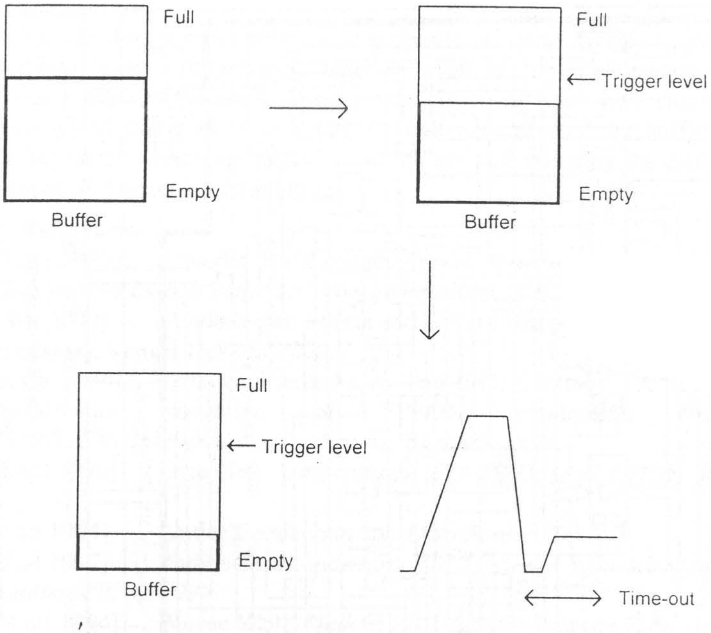

Figure 3 Improvements of the syncronization procedure

Figure 3 shows the architecture of an EPROM emulator which is based on a microcontroller and an ASIC. The emulator is linked to a personal computer (PC) via a RS-232 interface. The target microcontroller can be seen as well. The interaction between the emulator's microcontroller and the target system is organized by two flags and two registers: Read Register (RR) and Write Register (WR). A set flag R is an indication that the target microcontroller has written a byte in the RR register and it should be read from the emulator's microcontroller. When the byte is read, the flag R is cleared automatically. A set flag W signals that the emulator's microcontroller has moved a byte to the WR register and it should be read by the target microcontroller. The architecture provides possibility the target microcontroller to clear the flag W by generating a specific address (7FFDH).

The ASIC has been implemented by two FPGAs ispLSI 1016 produced by Lattice. A hardware description language called ABEL was used as a design entry language. The Synario CAD tool was employed for logic optimization, mapping and routing.

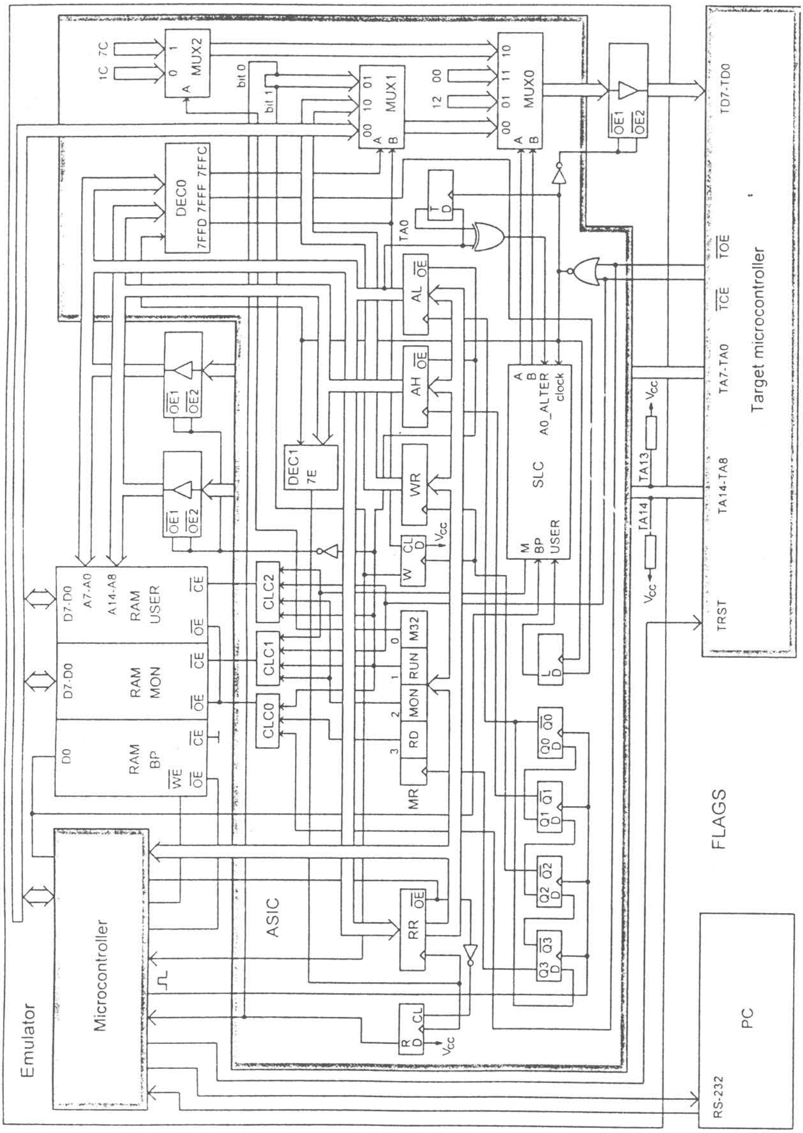

Figure 4 The EPROM emulator example

4. Conclusions

While the design process for some embedded systems is truly formalized and supported by relevant software tools, there are applications which are too specific to be designed completely by tools. The communication between microcontrollers and ASIC belongs to this case. The practical implementation of the buffers and the synchronisation procedure are decisions which are still taken by the designers and may be supported by tools in the future.

5. References

[Kara 1999a] Zdravko Karakehayov, Knud Christensen and Ole Winther, *Embedded Systems Design with 8051 Microcontrollers*, Marcel Dekker, 1999.

[Srir 2000] Sundararajan Sriram and Shuvra Bhattachacharyya, *Embedded Multiprocessors*, Marcel Dekker, 2000.

[Kara 1999b] Zdravko Karakehayov and Emil Saramov, "A case study in hardware-software co-design of EPROM emulators", *Proceedings ELECTRONICS'99 International Conference,* Sozopol, 1999.

[Data 1996] Data I/O Corporation, *SYNARIO, ABEL-HDL Reference*, 1996.

[Latt 1994] Lattice Semiconductor, *Data Book*, 1994.

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

[Wolf 1994] Wayne Wolf, *Modern VLSI Design*, Prentice Hall, 1994.

[Arms 1993] James R. Armstrong and F.Gail Gray, *Structured Logic Design with VHDL*, PTR Prentice Hall, 1993.

[Gajs 1994] Daniel D. Gajski, Frank Vahid, Sanjiv Narayan and Jie Gong, *Specification and Design of Embedded Systems*, PTR Prentice Hall, 1994.