RISC Microprocessor for a Wireless Communication System in Single Chip

Ricardo Linder, Geraldo Benício, José C. da Costa, Adson Rocha, Leonardo Menezes
linderrr@bigfoot.com
geraldob@anatel.gov.br
camargo@ene.unb.br
adson@ene.unb.br
leonardo@ene.unb.br

Universidade de Brasília – UnB
Departamento de Engenharia Elétrica – LPCI (Lab. de Proj. de Circuitos Integrados)
Caixa Postal 4386 – Brasília – DF – 70919-970

Abstract

The design of a sixteen bits RISC Microprocessor for a wireless communication an data processing system on a single chip, is presented emphasizing digital hardware and basic software.

1 Introduction

In this work the processor description of a wireless communication and data processing system, implemented in a single chip, is proposed. It is composed by a digital processor and memory bank, digital and analog interface and RF transceiver, all realized on a single monolithic integrated circuit. With such an approach power and area efficiency enhancements as well as better dynamic performances are expected in comparison to systems employing several chips [MAC 97]. The system employs a 0.8 micron double-poly, double metal CMOS technology (from AMS), which is adequate for the proposed 900 MHz RF transceiver and the 200 MHz processor implementation [IWA 96].

2 Processor Description

A RISC architecture [MUR 96] [DOL 97] [MON 96] was adopted for the processor. The characteristics are: simple instructions set, instructions with the same size, majority instructions are executed in only one cycle time, fast processing speed, simplest hardware, shortest design cycle. These characteristics allow the implementation of a simple and fast hardware, expending a minor area of chip. The processor's block diagram is shown on Fig.1.

3 Architecture Specification

The project has restrictions like limited area and low power supply, which are the basic parameters to list the specifications. This processor has a small layout and a simple instruction set. These factors help us to choose the RISC methodology. The word and the register quantities must be reasonable in size to make the application development possible without a large area or a low performance (small transference rate). The power supply must

to limit the power dissipation on chip, but not decreasing the circuit speed until a level below of the one required.

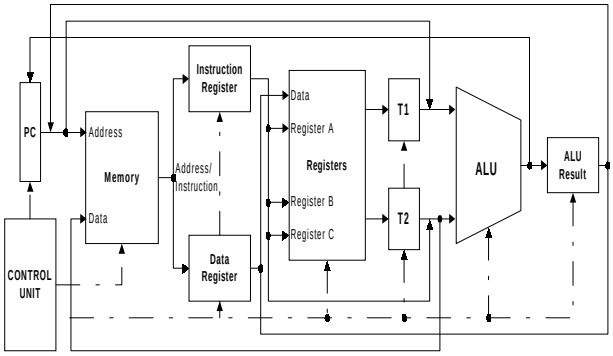

Figure 1 - Processor´s block diagram.

The specification described below intend to obey the project requirements without great complexity:

- RISC Processor, 16-b and 200 MHz clock;
- 3.3 V Power Supply;
- 100mW power consume;
- Fixed Point Arithmetic Logic Unit (ALU);
- Sixteen registers in one Bank;
- 8KB Memory Unit (SRAM);

The ALU was designed for fixed-point operations without multiplication by hardware.

4 Microinstructions

The microinstruction set should be able to execute all operations demanded by the application [PAT 76] [THO 85] [RAM 74]. On other hand, the complexity of this chip and it's cost, which are relative to area of IC, define the division between the hardware and the software.

Keeping in mind the principle of simplicity, the microinstruction set has to have the smallest possible number of formats. After critical analysis the microinstruction set can have tree possible formats:

Type R

Operation	Source Reg. 1	Source Reg. 2	Destination Reg.
4 bits	4bits	4 bits	4bits

Type I

Operation	Source Reg. 1	Source Reg. 2	Offset
4 bits	4bits	4 bits	4bits

Type J

Operation	Source Reg.	Address
4 bits	4bits	8 bits

Figure 2 - Instruction's formats.

The largest difficulty in defining the microinstruction set resides in the fact of that instruction of type R need four fields:

1. Operation - that contains the code that identifies the instruction;
2. Source Register - that identifies the source register. As there are sixteen registers, four bits will be obligatorily allocated to identify them;
3. Source Register 2 - that identifies the second source register. Its size is, in the same way, of four bits;
4. Destination Register - that identifies the destination register. Its size is, for the same reason of the above two, of four bits.

Therefore, three of the fields have a defined size, obligatorily, as being of four bits. This configuration consumes twelve bits and, to project a set of instructions of sixteen bits, the field " Operation " is thus limited to the use of only four bits for its code. With only four bits to identify the instruction, the set is reduced the possibility of creating a universe of sixteen instructions.

Type I instructions have only four bits, for the field "Offset". Typically this field stores a constant value equivalent to the Offset, using the PC that indicates a base address.

In type J instructions remaining only eight bits for the field "Address". The instruction thus can access a register with a base address of sixteen bits and supply Offset using the eight bits. This means an address capacity of 64B with Offset of 255B, which is more than the necessary for UnB2000-1, since it possesses a memory of 8KB.

The selection of the microinstructions to be used, takes in consideration the project objectives, the limitations of the architecture and, mainly, the needs of the application.

To create a generic set, it is necessary to cover four basic categories: Arithmetic, Logic, Transfer and Branch.

5 Pseudoinstructions

Sixteen microinstructions are insufficient to implement efficient applications; therefore it is necessary to expand by implementing a pseudoinstruction group.

Each pseudoinstruction, actually, corresponds to one microinstruction or a sequence of microinstructions, forming a new and more complex instruction.

Since the size of the instructions, in the microinstruction level, does not reach beyond a tangible limit, the project Unb2000-1 will use thirty-two bits as the size of each pseudoinstruction. This option allows to implement a larger pseudoinstructions number and with a larger number of parameters.

Not all of the pseudoinstructions have the same size. The freedom obtained in this level allows the use of larger instructions, which in this project will be of sixty-four bits. A larger size will be necessary to assist the specifications of some pseudoinstructions.

The assembler is the software that will convert that abstraction of a instruction set of thirty two bits in the UnB2000-1 set of sixteen instructions of sixteen bits.

6 Registers and Memory

By defining the number of registers as being sixteen, the next step is to decide their use.

The select choice was dedicate to each register a purpose, maintaining a small group with relative flexibility to be used in the storage of data. The registers dedicated are in principle those of control and will act as reference for several microinstructions.

The system has one memory bank used to save data and instructions with total size of 8KB (SRAM). This choice was done due the restrictions of the project, like area and consume, and the requisites to implement the applications suggested on the introduction, which demand, mainly, speed. The cell memory chooses uses current-mode write technique. Each cell memory has 7 transistors (7T), one more than 6T cell. That should be bigger than last one, but it's faster and has lower power consume [WAN 00] [HAD 00].

7 Conclusion

A RISC Microprocessor for a wireless communication emphasizing digital hardware and basic software was presented. The design methodology will be validated through realistic applications like telemetry and node for wireless LAN.

8 References

[PAT 76] PATTERSON, D. A. "STRUM: structured microprogram development system for correct firmware", ***IEEE Transactions on Computers***, vol. c-25 (10), pp. 974-985, Out. 1976.

[MAC 97] MACSHANE, E. A & SHENAI, K. "Functionally Integrated Systems on a Chip: Technologies, Architectures, CAD Tools and Applications" ***IEEE 1997 Innovative Architecture for Future Generation High Performance Processors***, pp. 67-75.

[MUR 96] MURABAYASHI, F. & SAWAMOTO, H. "2.5 V CMOS Circuit Techniques for a 200 MHz Superscalar RISC Processor". ***IEEE Journal of Solid State Circuits***, vol. 31, pp. 972-980, Jul. 1996.

[IWA 96] IWAI, H. "CMOS Technology for RF Application", ***Proc. Int. Conf. On Microelectronics***, Serbia, vol.1 pp. 14-17 2000.

[WAN 00] WANG, J. S.; TSENG, W.; HUNG-Yu Li, "Low power embedded SRAM with current-mode write technique", ***IEEE Journal of Solid State Circuits***, vol. 35, pp. 119 – 124, Jan. 2000.

[DOL 97] DOLLE, M. & JHAND, S. "A 32-b RISC/DSP Microprocessor with Reduced Complexity". ***IEEE Journal of Solid State Circuits***, vol. 32, Jul. 1997.

[THO 85] THOR, M. & SZTURMOWICZ, M. "Interrupts in modular microprogramming", ***Microprocessing and Microprogramming***, vol. 16 (4-5), pp. 325-330, 1985.

[MON 96] MONTANARO, R. T. W. & ANNE, K. "A 160 MHz, 32b, 0,5 W CMOS RISC Microprocessor". ***IEEE Journal of Solid State Circuits***, vol. 31, Nov. 1996.

[HAD 00] HADDAD, S. A. P. "Desenvolvimento de Células Digitais CMOS e Células RF para um Sistema de Comunicação Integrado (Relatório da Disciplina Estágio Supervisionado II)". Universidade de Brasília, 2000.

[RAM 74] RAMAMOORTHY, V. & TSUCHIYA, M. "A high-level language for horizontal microprogramming", ***IEEE Transactions on Computers***, vol. c-25 (8), pp. 791-801, Aug. 1974.