USP - ICMC - SSC
SCE 0703 (PISE) - 2o. Semestre 2008

Disciplina de Projeto e Implementação de Sistemas Embarcados I

Prof. Fernando Santos Osório
Email: fosorio [at] { icmc. usp. br , gmail. com }
Web: http://www.icmc.usp.br/~fosorio/

Aula 09

Aula 09 – Processador NIOS II

Temas Abordados:

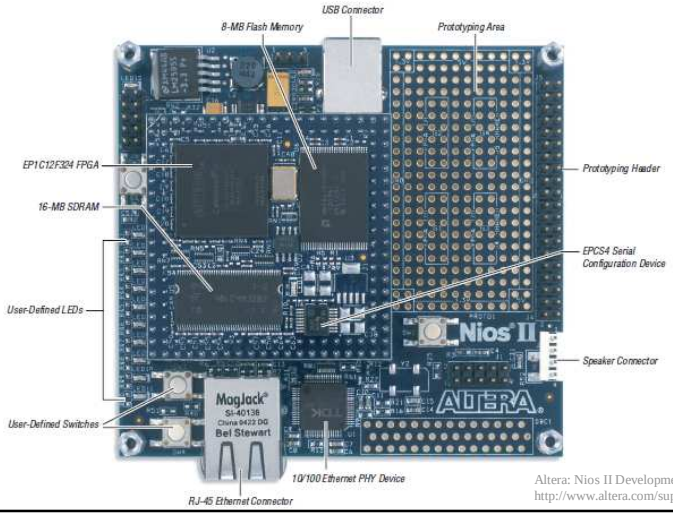
1. **Processador NIOS II**

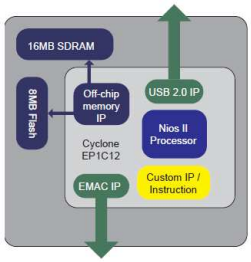
2. **Simulação do NIOS II**
   - > **NiosII-IIS: NiosII Instruction Set Simulator**
   - > **NiosII IDE**

3. **Programação do NIOS II**

Processador NIOS II

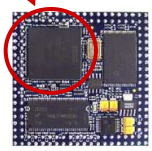
Nios II Evaluation Board





Firefly modules can be designed to fit any number of embedded device requirements.

CYCLONE FPGA EP1C12F324C8



Firefly module shown actual size (2" x 2")

Altera: Nios II Development Kit, Cyclone Edition
http://www.altera.com/support/ip/processors/kits/ips-nios2-dev-kits.html

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

Processador NIOS II

Nios II Processor - NiosII Evaluation Board

Nios II is a 32-bit embedded-processor architecture designed specifically for the Altera family of FPGA.

Nios II incorporates many enhancements over the original Nios architecture, making it more suitable for a wider range of embedded computing applications, from DSP to system-control.

Nios II architecture is a RISC soft-core architecture which is implemented entirely in the programmable logic and memory blocks of Altera FPGAs.

Altera: Nios II Development Kit, Cyclone Edition
http://www.altera.com/support/ip/processors/kits/ips-nios2-dev-kits.html

Processador NIOS II

Nios II Processor - NiosII Evaluation Board

Nios II is a 32-bit embedded-processor architecture designed specifically for the Altera family of FPGA.

Nios II incorporates many enhancements over the original Nios architecture, making it more suitable for a wider range of embedded computing applications, from DSP to system-control.

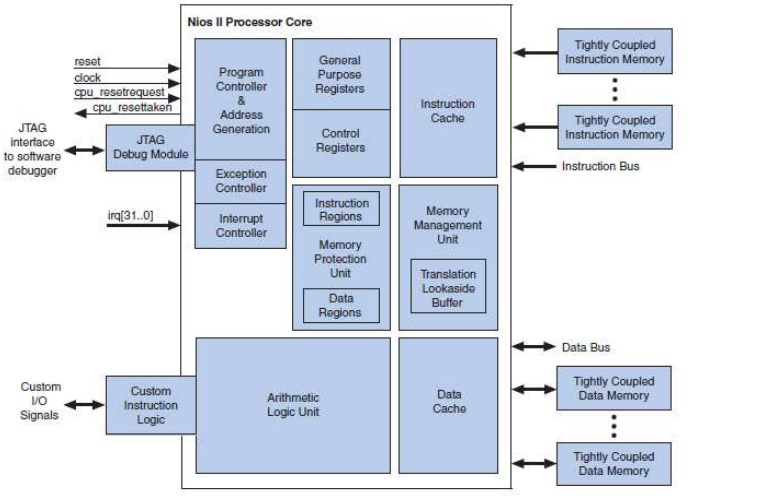
Nios II architecture is a RISC soft-core architecture which is implemented entirely in the programmable logic and memory blocks of Altera FPGAs.

Altera: Nios II Development Kit, Cyclone Edition
http://www.altera.com/support/ip/processors/kits/ips-nios2-dev-kits.html

Processador NIOS II

Nios II Processor Architecture

Nios II Processor Core Block Diagram

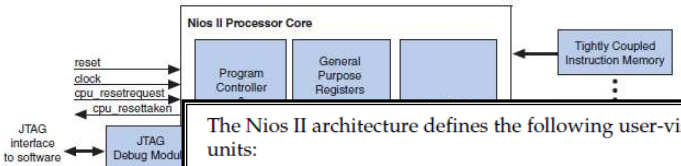


Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

Processador NIOS II

Nios II Processor Architecture

Nios II Processor Core Block Diagram



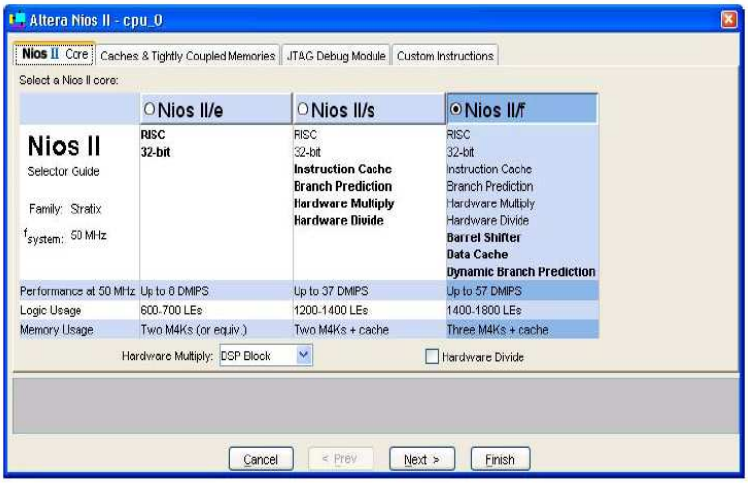
The Nios II architecture defines the following user-visible functional units:

- Register file
- Arithmetic logic unit (ALU)
- Interface to custom instruction logic
- Exception controller
- Interrupt controller
- Instruction bus
- Data bus
- Memory management unit (MMU)
- Memory protection unit (MPU)
- Instruction and data cache memories
- Tightly-coupled memory interfaces for instructions and data
- JTAG debug module

Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

Processador NIOS II

Nios II Processor Architecture



Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

Processador NIOS II

Nios II Processor Architecture

Table 5–1. Nios II Processor Cores (Part 1 of 2)

Feature		Core		
		Nios II/e	Nios II/s	Nios II/f
Objective		Minimal core size	Small core size	Fast execution speed
Performance	DMIPS/MHz *(1)*	0.15	0.74	1.16
	Max. DMIPS *(2)*	31	127	218
	Max. f_{MAX} *(2)*	200 MHz	165 MHz	185 MHz
Area		< 700 LEs; < 350 ALMs	< 1400 LEs; < 700 ALMs	< 1800 LEs; < 900 ALMs
Pipeline		1 Stage	5 Stages	6 Stages
External Address Space		2 Gbytes	2 GBytes	2 GBytes
Instruction Bus	Cache	–	512 bytes to 64 kbytes	512 bytes to 64 kbytes
	Pipelined Memory Access	–	Yes	Yes
	Branch Prediction	–	Static	Dynamic
	Tightly Coupled Memory	–	Optional	Optional

Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

Processador NIOS II

Nios II Processor Architecture

Table 5–1. Nios II Processor Cores (Part 1 of 2)

Feature		Core		
		Nios II/e	Nios II/s	Nios II/f
Objective		Minimal core size	Small core size	Fast execution speed
Performance	DMIPS/MHz *(1)*	0.15	0.74	1.16
	Max. DMIPS *(2)*	31	127	218
	Max. f_{MAX} *(2)*	200 MHz	165 MHz	185 MHz
Area		< 700 LEs; < 350 ALMs	< 1400 LEs; < 700 ALMs	< 1800 LEs; < 900 ALMs

Table 1. Cyclone Family Device Overview

Device	Logic Elements	Maximum PLLs	M4K RAM Blocks	Total RAM Bits	Maximum User I/O Pins
EP1C3	2,910	1	13	59,904	104
EP1C4	4,000	2	17	78,336	301
EP1C6	5,980	2	20	92,160	185
EP1C12	12,060	2	52	239,616	249
EP1C20	20,060	2	64	294,912	301

Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

Processador NIOS II

Nios II Processor Architecture

Table 5–1. Nios II Processor Cores (Part 1 of 2)

Feature		Core		
		Nios II/e	Nios II/s	Nios II/f
Objective		Minimal core size	Small core size	Fast execution speed
Performance	DMIPS/MHz (1)	0.15	0.74	1.16
	Max. DMIPS (2)	31	127	218
	Max. f_{MAX} (2)	200 MHz	165 MHz	185 MHz
Area		< 700 LEs; < 350 ALMs	< 1400 LEs; < 700 ALMs	< 1800 LEs; < 900 ALMs

Table 2. Comparison of FPGAs in the Stratix Series

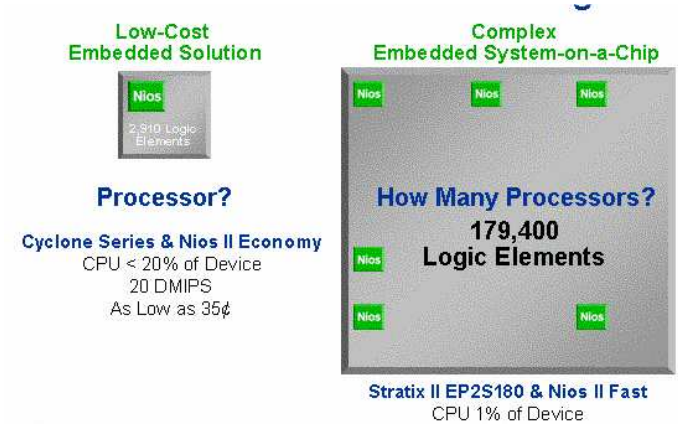
Feature	Stratix IV E	Stratix III L	Stratix III E	Stratix II	Stratix
Equivalent Logic Elements	105,600 to 681,100	47,500 to 338,000	47,500 to 254,400	15,600 to 179,400	10,570 to 79,040
Adaptive Logic Modules	42,240 to 272,440	19,000 to 135,200	19,000 to 101,760	6,240 to 71,760	N/A
Total RAM (Kbits)	8,244 to 22,977	1,836 to 16,272	5,328 to 14,688	410 to 9,163	899 to 7,253

Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

USP - SCE-703 - 2008-2
Proj. e Implem. de Sistemas Embarcados I
Prof. Fernando Osório

Processador NIOS II

Nios II Processor Architecture



Another common representation of the Dhrystone benchmark is the DMIPS - Dhrystone MIPS:
obtained when the Dhrystone score is divided by 1,757 (Dhrystone is a synthetic computing benchmark program)

Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

Processador NIOS II

Nios II Processor Architecture

- Soft IP Core (SOPC - System on a Programable Chip)
  A *soft-core processor* is a microprocessor fully described in software, usually in an HDL, which can be synthesized in programmable hardware, such as FPGAs.
- Reduced Instruction Set Computer (RISC)
- No pipeline, 5 or 6 stages pipeline configurations
- Full 32-bit instruction set, data path, and address space
- 32 general-purpose registers: r0 - r31
- 6 control registers
- 32 external interrupt sources
- Access to a variety of on-chip peripherals, and interfaces to off-chip memories and peripherals
- Software development environment based on the GNU C/C++ tool chain and Eclipse IDE

Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

Processador NIOS II

Nios II Processor Architecture

Register File

The Nios II architecture supports a flat register file, consisting of thirty two 32-bit general-purpose integer registers, and up to thirty two 32-bit control registers. The architecture supports supervisor and user modes that allow system code to protect the control registers from errant applications.

General Purpose Registers

□> The Nios II has 32 regs of 32-bit general-purpose: ***Registers r0-r31***
r0 (zero) has always the value 0 (zero-register)
□ r31 (ra) holds the return address used by procedure calls and is implicitly accessed by call and ret instructions
□> C and C++ compilers use a common procedure-call convention, assigning a special meaning to register r1through r23 and r26 to r28 (Application Binary Interface)
> Access to certain registers, such as et(r24), bt(r25), ea(r29) and ba(r30) is limited to certain execution modes

Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

Nios II Processor Architecture

Control Registers

- There are six control registers: **ctl0 to ctl5**
- Control registers can only be accessed in supervisor mode by the special instructions rdctl and wrctl
- Each register has also an individual name that is recognized by the assembler: status, estatus, bstatus, ienable, ipending, cpuid

Operation Modes

- Nios II is prepared to operate in the following modes
  - * **Supervisor mode**: All defined processor functions are available and unrestricted
  - * **User mode**: Some processor features, like the control registers, are instructions accessing them will cause an exception
  - * **Debug mode**: Debug mode is used by software developing tools. In debug mode all functions are accessible

Altera: Processor Architecture, Nios II Processor Reference Handbook - File: n2cpu_nii51002.pdf

Nios II Processor Architecture

Nios II Addressing Modes

- The Nios II architecture supports the following addressing modes:
  - *Register addressing*: all operands are registers, e.g. `add r6, r7, r8`
  - *Displacement addressing*: address is calculated as the sum of a register and a signed, 16-bit immediate value, e.g. `ldw r6, 100(r5)`
  - *Immediate addressing*: the operand is a constant within the instruction itself, e.g. `movi r6, -30`
  - *Register indirect addressing*: as in displacement adressing, but the displacement is the constant 0, e.g. `ldw r6, (r5)`
  - *Absolute addressing*: as in displacement addressing, but register r0 (zero) is used, e.g. `ldw r6, 100(r0)`

Processador NIOS II

Nios II Processor Architecture

Nios II Instruction Set

Data Transfer

- Nios II has a RISC (Reduced Instruction Set Computer) load-store architecture
- Load and Store instructions handle all data movement between registers, memory, and peripherals
- There are instructions for both cached (`ldw`, `stw`) and uncached access (`ldwio`, `stwio`)
- There are instructions that support half-word and byte transfers
- *Example*: `stw r5, (r9)` stores the value of word in `r5` in the memory location that is given by `r9`

Processador NIOS II

Nios II Processor Architecture

Arithmetic Logic Unit

The Nios II ALU operates on data stored in general-purpose registers. ALU operations take one or two inputs from registers, and store a result back in a register. The ALU supports the data operations shown in Table 2–1.

Table 2–1. Operations Supported by the Nios II ALU

Category	Details
Arithmetic	The ALU supports addition, subtraction, multiplication, and division on signed and unsigned operands.
Relational	The ALU supports the equal, not-equal, greater-than-or-equal, and less-than relational operations (==, != >=, <) on signed and unsigned operands.
Logical	The ALU supports AND, OR, NOR, and XOR logical operations.
Shift and Rotate	The ALU supports shift and rotate operations, and can shift/rotate data by 0 to 31 bit-positions per instruction. The ALU supports arithmetic shift right and logical shift right/left. The ALU supports rotate left/right.

Processador NIOS II

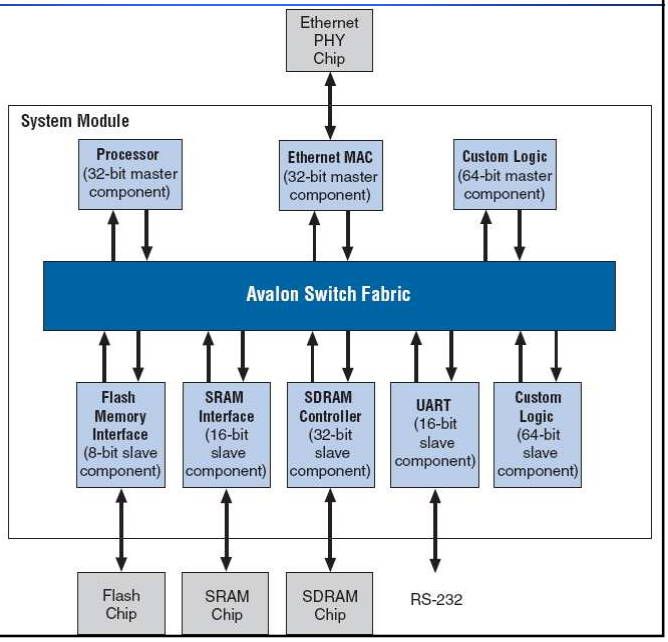
Nios II

Processor

Architecture

I/O

Memoria

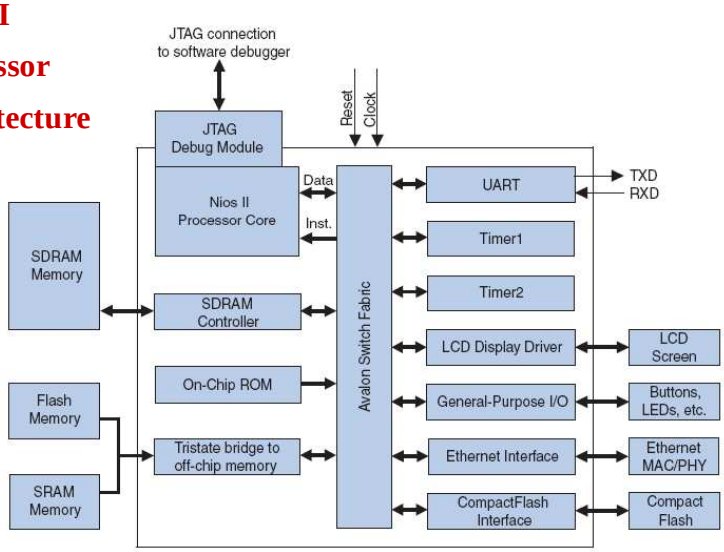


Processador NIOS II

Nios II

Processor

Architecture



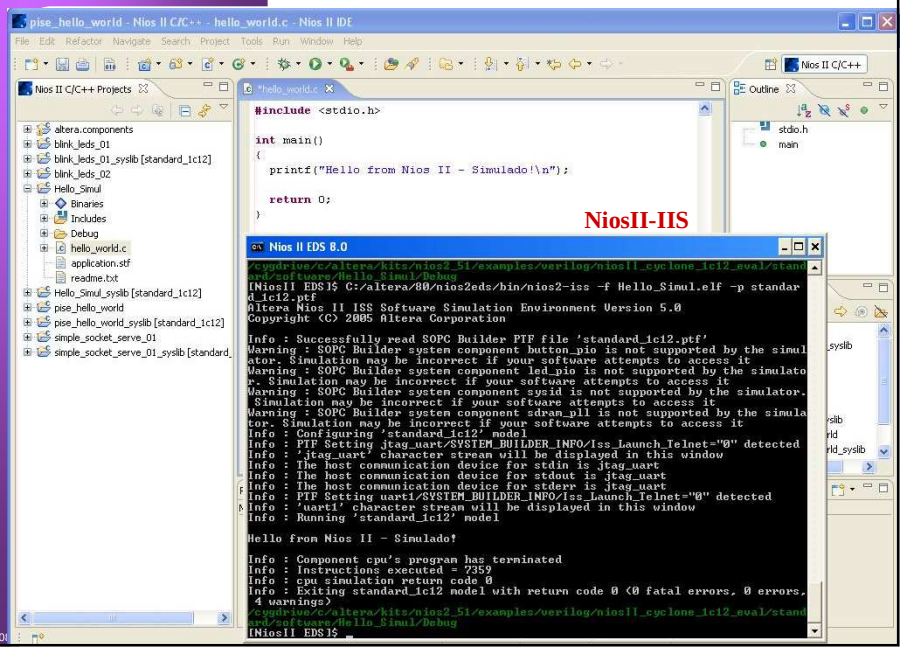
Processador NIOS II

Nios II Processor Architecture

Intruction Set - Example of Code

```
0x01009c68 <write+0x114>:   0xdf000717  ldw fp, 28(sp)          [memAddr=0x1fff8b4 ldData=0x1ffffe8 dstReg=fp]
0x01009c6c <write+0x118>:   0xdec00904  addi sp, sp, 36         [dstData=0x1fff8bc dstReg=sp]
0x01009c70 <write+0x11c>:   0xf800283a  ret                     [targetPcb=0x1006d68]
0x01006d68 <_write_r+0x24>: 0x1007883a  mov r3, r2              [dstData=0x11 dstReg=r3]
0x01006d6c <_write_r+0x28>: 0x00bfffc4  movi r2, -1             [dstData=0xffffffff dstReg=r2]
0x01006d70 <_write_r+0x2c>: 0x18800526  beq r3, r2,0x1006d88    [failed]
0x01006d74 <_write_r+0x30>: 0x1805883a  mov r2, r3              [dstData=0x11 dstReg=r2]
0x01006d78 <_write_r+0x34>: 0xdfc00117  ldw ra, 4(sp)           [memAddr=0x1fff8c0 ldData=0x1003f58 dstReg=ra]
0x01006d7c <_write_r+0x38>: 0xdc000017  ldw r16, 0(sp)          [memAddr=0x1fff8bc ldData=0x11 dstReg=r16]
0x01006d80 <_write_r+0x3c>: 0xdec00204  addi sp, sp, 8          [dstData=0x1fff8c4 dstReg=sp]
0x01006d84 <_write_r+0x40>: 0xf800283a  ret                     [targetPcb=0x1003f58]
0x01003f58 <fflush+0x74>:   0x80a1c83a  sub r16, r16, r2        [dstData=0x0 dstReg=r16]
0x01003f5c <fflush+0x78>:   0x90a5883a  add r18, r18, r2        [dstData=0x1fffb05 dstReg=r18]
0x01003f60 <fflush+0x7c>:   0x00bff716  blt r0, r2, 0x1003f40   [passed]
0x01003f40 <fflush+0x5c>:   0x04000d0e  bge r0, r16,0x1003f78   [passed]
0x01003f78 <fflush+0x94>:   0x0007883a  mov r3, r0              [dstData=0x0 dstReg=r3]
0x01003f7c <fflush+0x98>:   0x1805883a  mov r2, r3              [dstData=0x0 dstReg=r2]
0x01003f80 <fflush+0x9c>:   0xdfc00317  ldw ra, 12(sp)          [memAddr=0x1fff8d0 ldData=0x1000698 dstReg=ra]
0x01003f84 <fflush+0xa0>:   0xdc000217  ldw r16, 8(sp)          [memAddr=0x1fff8cc ldData=0x11 dstReg=r16]
0x01003f88 <fflush+0xa4>:   0xdc400117  ldw r17, 4(sp)          [memAddr=0x1fff8c8 ldData=0x1fffa98 dstReg=r17]
0x01003f8c <fflush+0xa8>:   0xdc800017  ldw r18, 0(sp)          [memAddr=0x1fff8c4 ldData=0xdeadbeef dstReg=r18]
0x01003f90 <fflush+0xac>:   0xdec00404  addi sp, sp, 16         [dstData=0x1fff8d4 dstReg=sp]
0x01003f94 <fflush+0xb0>:   0xf800283a  ret                     [targetPcb=0x1000698]
```


Simulação do NIOS II
NiosII-IIS: NiosII Instruction Set Simulator



Simulação do NIOS II

NiosII-IIS: NiosII Instruction Set Simulator

Usando o NiosII-IIS:

- Gerar um programa compilado usando o NIOS II IDE

- Abrir o NiosII Command Shell (Terminal)
- Executar o NiosII-ISS.exe
Diretório:
C:\altera\80\nios2eds\bin\nios2-iss.exe

- Exemplo de Comando:
C:/altera/80/nios2eds/bin/nios2-iss -f Meu_Prog.elf -p standard_1c12.ptf

C:/altera/80/nios2eds/bin/nios2-iss -td -f Meu_Prog.elf -p standard_1c12.ptf
-tr
-h

Simulação do NIOS II

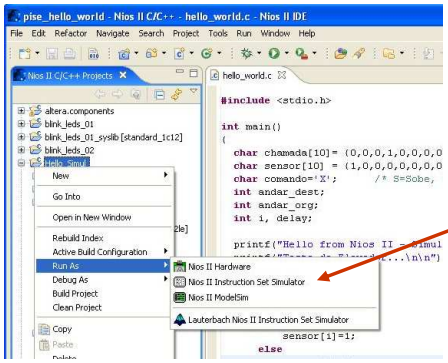
```
[NiosII EDS]$ C:/altera/80/nios2eds/bin/nios2-iss -f Hello_Simul.elf -p standard_1c12.ptf

Altera Nios II ISS Software Simulation Environment Version 5.0
Copyright (C) 2005 Altera Corporation

Info : Successfully read SOPC Builder PTF file 'standard_1c12.ptf'
Warning : SOPC Builder system component button_pio is not supported by the simulator.
Simulation may be incorrect if your software attempts to access it
Warning : SOPC Builder system component led_pio is not supported by the simulator.
Simulation may be incorrect if your software attempts to access it
Warning : SOPC Builder system component sysid is not supported by the simulator.
Simulation may be incorrect if your software attempts to access it
Warning : SOPC Builder system component sdram_pll is not supported by the simulator.
Simulation may be incorrect if your software attempts to access it
Info : Configuring 'standard_1c12' model
Info : PTF Setting jtag_uart/SYSTEM_BUILDER_INFO/Iss_Launch_Telnet="0" detected
Info : 'jtag_uart' character stream will be displayed in this window
Info : The host communication device for stdin is jtag_uart
Info : The host communication device for stdout is jtag_uart
Info : The host communication device for stderr is jtag_uart
Info : PTF Setting uart1/SYSTEM_BUILDER_INFO/Iss_Launch_Telnet="0" detected
Info : 'uart1' character stream will be displayed in this window
Info : Running 'standard_1c12' model

Hello from Nios II - Simulado!

Info : Component cpu's program has terminated
Info : Instructions executed = 7359
Info : cpu simulation return code 0
Info : Exiting standard_1c12 model with return code 0 (0 fatal errors, 0 errors, 4 warnings)
/cygdrive/c/altera/kits/nios2_51/examples/verilog/niosII_cyclone_1c12_eval/standard/software/Hello_Simul/Debug

[NiosII EDS]$
```

Simulação do NIOS II

NiosII-IIS: NiosII Instruction Set Simulator

Usando o NiosII-IIS com o NIOS II IDE:

- Configurar o PROJECT para que seja usado o NiosII-IIS
> Clicar com o botão da direita
> Selecionar uso do simulador

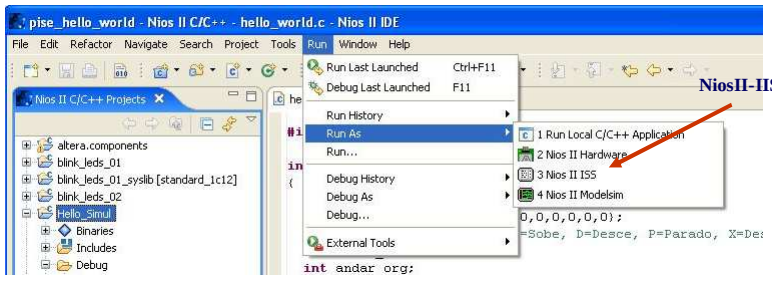


Simulação do NIOS II

NiosII-IIS: NiosII Instruction Set Simulator

Usando o NiosII-IIS com o NIOS II IDE:

- Configurar o PROJECT para que seja usado o NiosII-IIS
> Clicar com o botão da direita
> Selecionar uso do simulador



INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO
Web institucional: http://www.icmc.usp.br/ssc/
Página pessoal: http://www.icmc.usp.br/~fosorio/
E-mail: fosorio [at] icmc. usp. br ou fosorio [at] gmail. com

Disciplina de Proj. e Implementação de Sistemas Embarcados I
Ferramentas: Altera - Quartus, NIOS II, Cyclone Dev-Kit
Web disciplina: Http://www.icmc.usp.br/~fosorio/
> Programa, Material de Aulas, Critérios de Avaliação,
> Material de Apoio, Trabalhos Práticos