

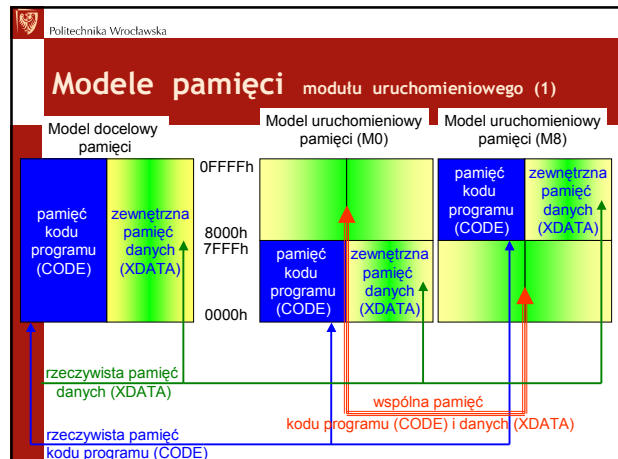
# Politechnika Wrocławska


## Technika cyfrowa 2

### wykład 14

Evaluation Kit  
Perspektywy rozwoju mikrokontrolerów i nie tylko

Katedra Metrologii Elektronicznej i Fotonicznej  
Andrzej Stępień



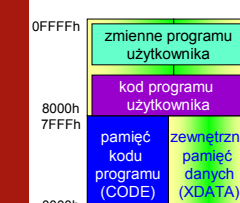



## Modele pamięci modułu uruchomieniowego (2)

Model uruchomieniowy pamięci (M0)

| XSEG AT 0C000h |                    |
|----------------|--------------------|
| wart_1:        | DS 1 ; rezerwacja  |
| wart_2:        | DS 14 ; zmiennych  |
| wart_3:        | DS 5 ; użytkownika |

| CSEG AT 8000h    |                           |
|------------------|---------------------------|
| Prog_Pocz:       | JMP Dalej_Prog_Pocz       |
| CD_INT0:         | ORG 8003h ; obsługa INT0# |
|                  | RETI                      |
| CD_ADC:          | ORG 8043h ; obsługa ADC   |
|                  | RETI                      |
| Dalej_Prog_Pocz: | ; program główny          |



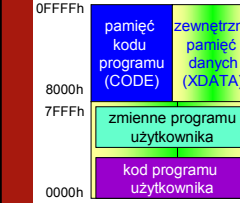


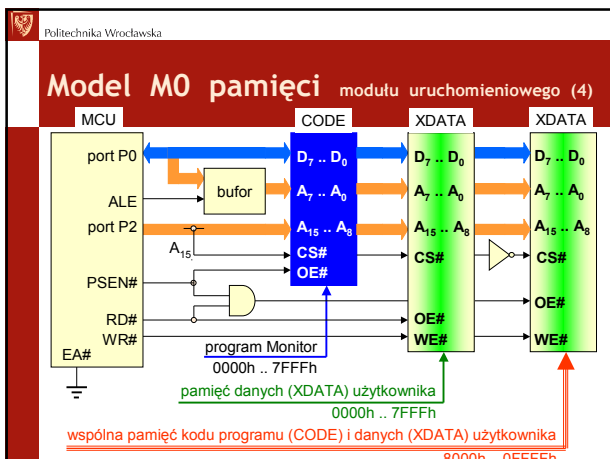
## Modele pamięci modułu uruchomieniowego (3)


Model uruchomieniowy pamięci (M8)

| XSEG AT 4000h |                    |
|---------------|--------------------|
| wart_1:       | DS 1 ; rezerwacja  |
| wart_2:       | DS 14 ; zmiennych  |
| wart_3:       | DS 5 ; użytkownika |

| CSEG AT 0000h    |                           |
|------------------|---------------------------|
| Prog_Pocz:       | JMP Dalej_Prog_Pocz       |
| CD_INT0:         | ORG 0003h ; obsługa INT0# |
|                  | RETI                      |
| CD_ADC:          | ORG 0043h ; obsługa ADC   |
|                  | RETI                      |
| Dalej_Prog_Pocz: | ; program główny          |







## Monitor - Keil Software (1)

INSTALL PROCEDURE FOR MONITOR-51 V2.9  
COPYRIGHT KEIL ELEKTRONIK GmbH 1988-2000

INSTALL serialtype [xdatastart] [codestart] [PROMCHECK][BANK]]]

serialtype=0 using TIMER 1 9600 bps at 11.059 MHz CPU Clock  
1 using baudr. gen. 9600 bps at 12.000 MHz (80515/80517)  
2 using TIMER 2 9600 bps at 12.000 MHz CPU Clock  
3 using serial interface 1 9600 bps at 12.000 MHz (80517)  
4 using T2 9600 bps at 12 MHz for DALLAS 80C320/520/530  
5 using Ser.Ch.1 9600 bps at 12 MHz (DALLAS 80C320/520/530)  
6 using external UART 16450/16550  
7 using TIMER 1 with self adjusting baudrate  
8 using TIMER 2 with self adjusting baudrate  
9 using baudr. gen. with self adjusting baudrate  
10 using serial interface 1 with self adj. bdr. for 80517(A)  
11 using TIMER 2 with self adj. bdr. for DALLAS 80C320/520/530  
12 using Ser.Ch.1 with self adj. bdr. for DALLAS 80C320/520/530

Politechnika Wroclawska

## Monitor - Keil Software (2)

**INSTALL** serialtype [xdatastart [codestart [PROMCHECK][BANK]]]

**xdatastart** must be a page-no. between 0 and 0FFH inclusive (using 256 byte)

**codestart** must be a block-no. between 0 and 0ECH inclusive (using 5 byte)

**PROMCHECK** checks whether there is a PROM or RAM at address 0. If the Monitor is created with the option PROMCHECK, the Monitor-51 checks on CPU reset if an EPROM or a RAM is present at code address 0. If an EPROM is detected, a JMP 0 instruction is executed that starts the code in the EPROM. PROMCHECK should be specified if the Monitor-51 code remains in the target system after the application has been programmed into an EPROM.

**BANK** generates Monitor-51 with banking support

EXAMPLE: INSTALL 0 7F 80

Politechnika Wroclawska

## Debugger - Keil Software (1)

*Getting Started and Creating Applications with  $\mu$ Vision2 and the C51 Microcontroller Development Tools User's Guide 06.2000*  
*Keil Elektronik GmbH and Keil Software, Inc.*

**Memory Commands:**

- ASM** Assembles in-line code.
- DEFINE** Defines typed symbols that you may use with  $\mu$ Vision2 debug functions.
- DISPLAY** Display the contents of memory.
- ENTER** Enters values into a specified memory area.
- EVALUATE** Evaluates an expression and outputs the results.
- MAP** Specifies access parameters for memory areas.
- UNASSEMBLE** Disassembles program memory.
- WATCHSET** Adds a watch variable to the Watch window.

Politechnika Wroclawska

## Debugger - Keil Software (2)

**Program Execution Commands:**

- Esc** Stops program execution.
- GO** Starts program execution.
- PSTEP** Steps over instructions but does not step into procedures or functions.
- OSTEP** Steps out of the current function.
- TSTEP** Steps over instructions and into functions.

**Simulate external I/O Devices**

External I/O devices are typical memory mapped. You may simulate such I/O devices with the **Memory Window** provided in the  $\mu$ Vision2 debugger. Since the C user program does not contain any variable declarations for such memory regions it is required that you map this memory with the MAP command:

**MAP X:0x1000, X:0x1FFF READ WRITE /\* MAP memory for I/O area \*/**

Politechnika Wroclawska

## Breakpoints - Keil Software

**Breakpoints**

You may use breakpoints in combination with debug functions to simulate the logic behind the I/O device. Example for a breakpoint definition:

**BS WRITE 0xFFE0, 1, "IO\_access ()"**

memory Access (Read, Write or both) ↑

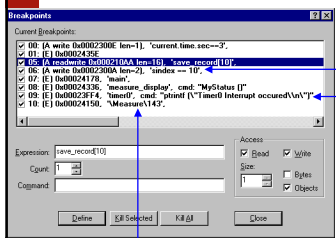
an address of a Conditional Break ↑

↑ a Command for a breakpoint

↑ number of times the breakpoint before the breakpoint is triggered.

Politechnika Wroclawska

## Breakpoints - Keil Software



Access Break (A) that halts program execution when the value **10** is written to the variable **sindex**.

Execution Break (E) that prints **T0 Interrupt occurred** in the **Output Window - Command** page when the target program reaches the timer0 function.

Execution Break (E) that halts when the target program reaches the code line 143 in the module MEASURE.

Politechnika Wroclawska

## ISD51 Breakpoints - Keil Software

- ISD51 (In-System Debugger) is a small debug monitor (500-700 bytes) that links with your 8051 target program. It interfaces to the  $\mu$ Vision Debugger using the 8051's on-chip UART and allows you to view memory, set breakpoints, single-step, and perform numerous debugging operations.
- If no breakpoints are set or if only hardware breakpoints or Flash breakpoints are set, the 8051 executes the user program at full speed.
- If software breakpoints are set, the 8051 enters the ISD51 interrupt function after each 8051 CPU instruction. The ISD51 interrupt checks to see if the 8051 program reached a breakpoint address and, if so, begins communication with the  $\mu$ Vision Debugger. 8051 programs execute considerably slower (about 100x) when software breakpoints are used.

Politechnika Wrocławska

## ISD51 Hardware Breakpoints - Keil Software

- Hardware Breakpoints use special on-chip breakpoint registers found on only a few devices. Microcontrollers with hardware breakpoint registers support only a few breakpoints (typically 1-4). But, they require almost no overhead and allow programs to execute at full speed. ISD51 currently supports hardware breakpoints on the TI MSC1210 devices.
- Known Issues - Following is a list of issues with hardware breakpoints on the MSC1210 devices:
  - Program execution may stop up to two CPU instructions after a breakpoint. This is a chip limitation and not a limitation of the software.
  - The **Read** and **Write** attributes are ignored for XDATA access breakpoints. Program execution stops on any access to the specified address.
  - Hardware breakpoints require use of the **interrupt 6** vector (address 0x33). Unfortunately, this interrupt is also used for other on-chip peripherals such as the A/D Converter, (milli)second timer, PFI, and SPI interface. ISD51 redirects interrupts for these peripherals to **interrupt 13** (address 0x6B). If your application uses peripherals that generate interrupt 6, you must change your interrupt service routine to use interrupt 13 instead.

Politechnika Wrocławska

## ISD51 Flash Breakpoints - Keil Software

- Flash Breakpoints use IAP (In-Application Programming) to replace breakpoint instructions with **CALL instructions** into the ISD51 code.
- Flash breakpoints are supported on all devices that offer IAP and have relatively small Flash block sizes (128 bytes or less). Numerous Flash breakpoints can be enabled simultaneously. They require almost no overhead and allow programs to execute at full speed.
- When you set a Flash breakpoint, ISD51 modifies the code memory and inserts a CALL instruction at the breakpoint address. When the breakpoint is reached, ISD51 is invoked (by the CALL). Run-time address checking (which is what software breakpoints do) is not required.
- To use Flash breakpoints, you must configure ISD51 for your target MCU. There is a place in the **ISD51.H** header file where you may configure the Flash block size and the algorithm to program a block.

Politechnika Wrocławska

## ISD51 Software Breakpoints - Keil Software

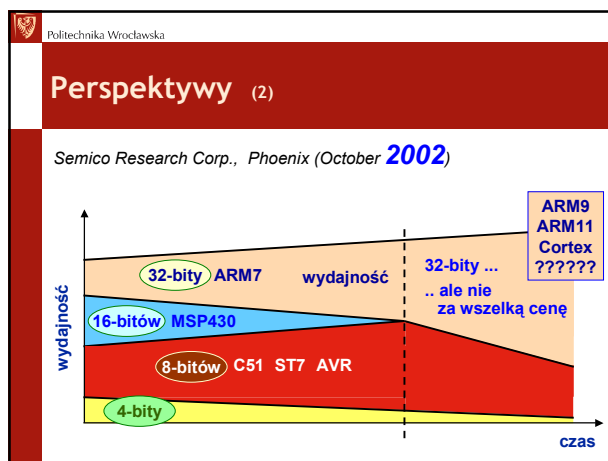
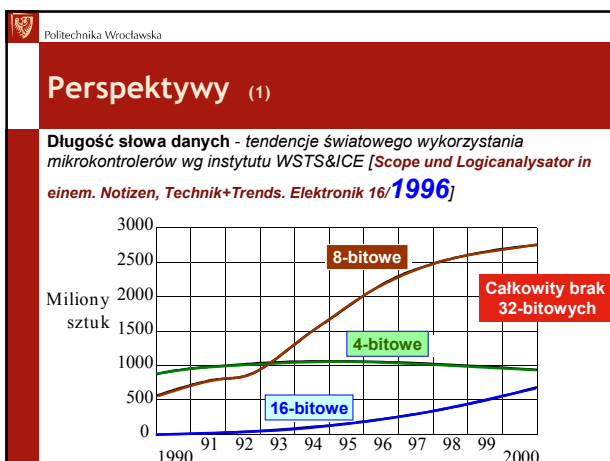
- ISD51 supports software breakpoints for all devices. When you set a software breakpoint, the breakpoint address is added to a table that ISD51 must check for each instruction that is executed. This check is performed in the ISD51 Interrupt.
- If ISD51 detects that a software breakpoint has been reached, it begins communication with the µVision Debugger.
- Note:
  - Software breakpoints in ISD51 cause programs to execute considerably **slower** (about 100x) than normal.
  - Software breakpoints and single-stepping do **not work in interrupt** service routines.

Politechnika Wrocławska

## ST7 Breakpoints - Software Interrupt - TRAP

**ST7. 8-BIT MCU FAMILY USER GUIDE. July 2002, STMicroelectronics.**


- Operation
  - PC = PC + 1
  - (SP--) = LSB (PC)
  - (SP--) = MSB (PC)
  - (SP--) = X
  - (SP--) = A
  - (SP--) = CC
  - PC = Vector Contents (FFFCh-FFFDh for ST723x4)**Op-Code = 1 byte**
- TRAP** produces the same effect as an externally-generated interrupt request, but under program control.
- When the user sets a breakpoint somewhere in the program, the debugger replaces the instruction at which the execution must stop with a **TRAP** instruction.
- RESET and **TRAP** are non-miscible interrupt.





Polttechnika Wroclawska

# Cortex Microcontrollers - www.arm.com (1)



# Politechnika Wrocławska

## Programmable Logic Device

- PLD (*Programmable Logic Device*)
- PAL (*Programmable Array Logic*)
- GAL (*Generic Array Logic*)
- FPGA (*Field Programmable Gate Array*)
- ASIC (*Application Specific Integrated Circuit*)

**OLMC Registered Configuration for Registered Mode**

- SYN=0
- AC0=1
- XOR=0 defines Active Low Output,
- XOR=1 defines Active High Output.
- AC1=0 defines this output configuration.
- Pin 1 controls common CLK for the registered outputs.
- Pin 13 controls common OE for the registered outputs.
- Pin 1 and Pin 13 are permanently configured as CLK & OE for registered output configuration.

**DIP (PLDCC) Package Pinouts**

**PROGRAMMABLE AND-ARRAY (64 X 32)**



Politechnika Wrocławska

## FPGA

**Configurable Logic Blocks (CLB)**

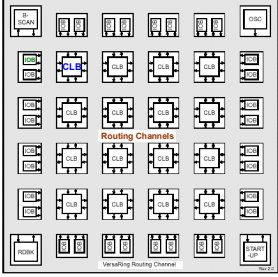
- Memory Look-Up Table
- AND-OR planes
- Simple gates

**Input / Output Blocks (IOB)**

- Bidirectional, latches, inverters, pullup/pulldowns

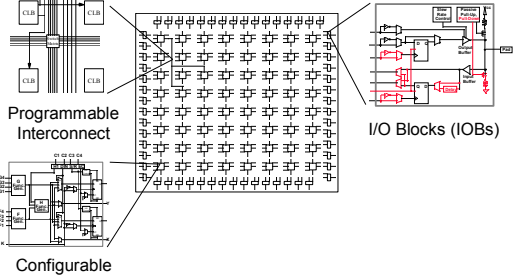
**Interconnect or Routing**

- Local, internal feedback, and global



Politechnika Wrocławska

## FPGA - Xilinx XC4000 Architecture



Programmable Interconnect

I/O Blocks (IOBs)

Configurable Logic Blocks (CLBs)

Politechnika Wrocławska

## FPGA - Xilinx Virtex-4 [www.xilinx.com](http://www.xilinx.com)

- XtremeDSP™ Slice, 18x18, two's complement, signed Multiplier, Built-In Accumulator (48-bits) & Adder/Subtractor
- High-speed memory interface support: DDR and DDR-2 SDRAM, QDR-II, RLDRAM-II, and FCRAM-II
- 1.2V core voltage, 1.5 to 3.3 V I/O Operation
- Built-In ChipSync™ Source-Synchronous Technology
- Digitally-controlled impedance (DCI) active termination
- RocketIO™ 622 Mb/s to 10+ Gb/s Multi-Gigabit Transceivers (MGT)
- IBM PowerPC RISC Processor Core (FX only) - PowerPC 405 (PPC405) Core - Auxiliary Processor Unit Interface (User Coprocessor)
- 960 Max User I/O
- Up to 200,000 logic cells including: up to 178,176 internal registers with clock enable (XC4VLX200), up to 178,176 look-up tables (LUTs), Logic expanding multiplexers and I/O registers
- JTAG support

Politechnika Wrocławska

## Into the future - [www.altium.com](http://www.altium.com)

- Microprocessors** created a revolution in electronics development because they allowed portions of the design problem to be moved into the highly fluid and easily updateable realm of software. Moving functionality into the 'soft' realm brings innumerable benefits to the design process. Critical design decisions can be made later, products can be brought to market earlier and then upgraded in the field, and functionality can be added to the product in software without adding to the overall product cost.
- The recent emergence of low-cost, high-capacity programmable devices such as **FPGAs** (Fieldprogrammable gate arrays) is redefining the boundaries between software and hardware. These devices allow the intelligent portions of the design to encompass not only software in the traditional sense, but the soft-wired blocks of hardware implemented inside the FPGA.