1. INTRODUCTION
2. CORE
3. ADDRESSING MODES
4. ASSEMBLY TOOLCHAIN
5. STVD7 DEBUGGER
6. HARDWARE TOOLS
7. PERIPHERALS
8. ST-REALIZER II
9. C TOOLCHAINS
STVD7: ST WINDOWS DEBUGGER FOR ST7 MCUs
ST7 Visual Debug Overview

- Hardware Development System Graphical Interface (32-bit)
- User Friendly Source Level Debugging
- Compatible with:
  - ST Macro-Assembler Tool Chain
  - Hiware/Metrowerks C Compiler Tool Chain
  - COSMIC C Compiler Tool Chain
- Complete ST7 Family Support
- Real Time Trace
- Breakpoints, Logic Analyser, Hardware Events...
- Version 3.x available: FULL IDE version
ST7 Visual Debug
Project settings

Workspace filename: C:\usr\STEPHIE\emu2b\adc\Exo4.wsp
Software Toolchain: ST7 Assembly Chain
Executable
Filename: exo4.s19
Build
Maker: ST7 Assembler Batch File
Maker filename:
Makefile or batch file: st_exo4.bat
"Build" command line: st_exo4.bat
"Rebuild All" command line: st_exo4.bat
Start "Build" / "Rebuild All" in: 

OK Cancel
ST7 Visual Debug
Micro Configuration
ST7 Visual Debug
Debugging Features

• Running the application
  ➢ Run
  ➢ Chip Reset
  ➢ Restart Application
  ➢ Continue
  ➢ Stop
  ➢ Step into
  ➢ Step over
  ➢ Step int ASM
  ➢ Step over Asm
  ➢ Step out
  ➢ Run to Cursor
  ➢ Set PC
ST7 Visual Debug
Debugging Features

• Common Debug Features
  ➢ Disassembly
  ➢ Memory Dump
  ➢ Instruction Breakpoints
  ➢ Data Breakpoints
  ➢ Watch
  ➢ Call Stack
  ➢ Local Variables
  ➢ ST7 Registers Window
  ➢ Symbol browser
  ➢ Peripheral registers window
Disassembly window

Location

Associated function and offset

Hexadecimal view

Disassembly

Symbolic Disassembly
ST7 registers

Two ways to interpret CC registers

Simulator-specific pseudo registers
Memory

Starting address of the dumped memory block

Binary Dump

ASCII dump
Instruction breakpoints

- Displays all instruction breakpoints (Active/Inactive)
- Enter counters and conditions
Quick Watch and Watch

- Evaluated expression
- Add the selected expression to the Watch window display
- Multiple tabs for Different contexts
- Select display format
Call stack

Double click here
Local variables

Call Stack

Auto detection of Local Variables and…

… function Parameters
ST7 Visual Debug
Debugging Features

- Peripheral registers window:
  - To visualise/ change peripheral's values
  - Accessing to some status register is not allowed

Bit-level description and control
Symbol browser

- Based on debug information
ST7 Visual Debug
Emulator-Specific functions
• BEM (Bus Event Machine)
ST7 Visual Debug
Emulator-Specific functions

- Performance Analysis (on EMU3):
ST7 Visual Debug
Emulator-Specific functions

- **Performance Analysis**: allows you to measure the time spent in a given portion of your code in order to evaluate the efficiency of your application.
### ST7 Visual Debug

**Emulator-Specific functions**

- **Trace:**
  - Choice of columns displayed free
  - Timestamp only present in EMU3
  - 1k Trace in EMU2B, 256k Trace in EMU3

<table>
<thead>
<tr>
<th>Record</th>
<th>Address</th>
<th>Memo...</th>
<th>Data</th>
<th>Event</th>
<th>Hexad...</th>
<th>Disassembly</th>
<th>Symbolic D...</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0x000005</td>
<td>In MCU</td>
<td>0xff</td>
<td>Data write</td>
<td></td>
<td></td>
<td></td>
<td>513 950 ns</td>
</tr>
<tr>
<td>17</td>
<td>0x00e20b</td>
<td>EMU</td>
<td>0x07</td>
<td>Operation code fetch</td>
<td>0x00b706</td>
<td>LD 0x06,A</td>
<td>LD PBOR,A</td>
<td>514 100 ns</td>
</tr>
<tr>
<td>18</td>
<td>0x00e20c</td>
<td>EMU</td>
<td>0x06</td>
<td>Operand fetch</td>
<td></td>
<td>LD PBOR,A</td>
<td></td>
<td>514 200 ns</td>
</tr>
<tr>
<td>19</td>
<td>0x000006</td>
<td>In MCU</td>
<td>0x00</td>
<td>Discarded</td>
<td></td>
<td></td>
<td></td>
<td>514 350 ns</td>
</tr>
<tr>
<td>20</td>
<td>0x000006</td>
<td>In MCU</td>
<td>0xff</td>
<td>Data write</td>
<td></td>
<td></td>
<td></td>
<td>514 450 ns</td>
</tr>
<tr>
<td>21</td>
<td>0x00e20d</td>
<td>EMU</td>
<td>0xa6</td>
<td>Operation code fetch</td>
<td>0x00a601</td>
<td>LD A,#0x01</td>
<td>LD A,#0x01</td>
<td>514 600 ns</td>
</tr>
<tr>
<td>22</td>
<td>0x00e20e</td>
<td>EMU</td>
<td>0x01</td>
<td>Operand fetch</td>
<td></td>
<td></td>
<td></td>
<td>514 700 ns</td>
</tr>
</tbody>
</table>
ST7 Visual Debug
Debugging Features

- GDB Console and Information window:
  - This window displays GDB output
  - Permits entry of commands
  - Permits to load binary file without debug information
Discrepancies

- Supported for:
  - EMU3, DVP3, ICD
- EMU3, DVP3
  - Describes the differences between emulated chip and your actual ST7
- ICD
  - Describes limitations that are chip-specific
Emulator update (EMU3)

- Emulators contain firmware that may be updated
- When the EMU3 is initialized
  - STVD7 checks the firmware versions
  - If needed, the user is prompted for automatic update
SIMULATOR OVERVIEW

• Scope
  ➢ Instruction level simulation
  ➢ Peripheral simulation: Port, Watchdog, Timers, ADC, SPI, I2C, SCI, E2PROM, WDG, MCC, RTC
  ➢ Pin level simulation (plotter)
  ➢ ST7 interrupt processing simulation
  ➢ To check overall code organization
  ➢ To verify simple peripheral configuration
  ➢ To check connection of interrupts to interrupt handlers

• Time management
  ➢ Display time
  ➢ break on time

• Breakpoints management
  ➢ Read & write access
  ➢ Stack Overflow/Underflow
  ➢ Invalid memory access detection

• No trace and no logic analyser features
Stack control

- Stop on stack overflow/underflow
- Configurable stack overflow
Read/write on the fly
Input pin stimulator

- Stimulate each pin with digital or analog value
Input pin stimulator features

- Trigger an analog or digital signal with or without a time delay
- Trigger a periodic digital signal with or without a time delay
  - Double click in value field
- Trigger a digital signal on the fly
  - Right click in value field
Stimuli file
Stimuli file syntax

- **pin** <pin name> -i <digital value> [time]
  Apply a digital value on an input pin
- **pin** <pin name> -a <analog value> [time]
  Apply an analog value on an input pin
- **pin** <pin name> -c <start digital value> [start time] <half period>
  Apply a square periodic signal on an input pin

Digital value is 0 or 1
Analog value is in the range [0-6.55]
Time is expressed in CPU cycles
Default time is current time
gdi pin command

• gdi pin -output_file <no/yes>
Default is no. If yes, pin input and output stimuli are logged in a file named port.out.

• gdi help pin
Get “gdi pin” syntax.

Stimuli pin commands (pin -i/-a/-c) used outside a stimuli file must be preceded by the gdi keyword.
The plotter

- Can plot the evolutions of global variables, pins, registers -> software oscilloscope!