

# Programování mikrokontrolerů

## JTAG

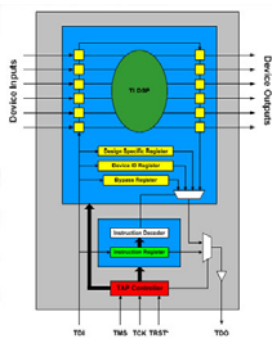


## What is JTAG

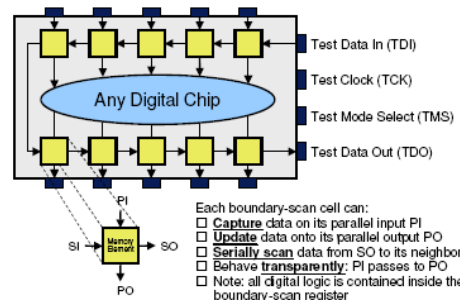
- Joint Test Action Group
- IEEE 1149.1 "Test Access Port and Boundary-Scan Architecture"



## Boundary-Scan



## Boundary-Scan



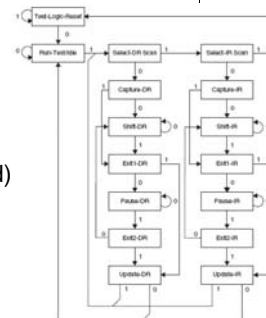
## JTAG and AVR

- PCB testing using boundary-scan capability
- programming non-volatile memory (flash, EEPROM)
- programming fuses and lock bits
- on-chip debugging



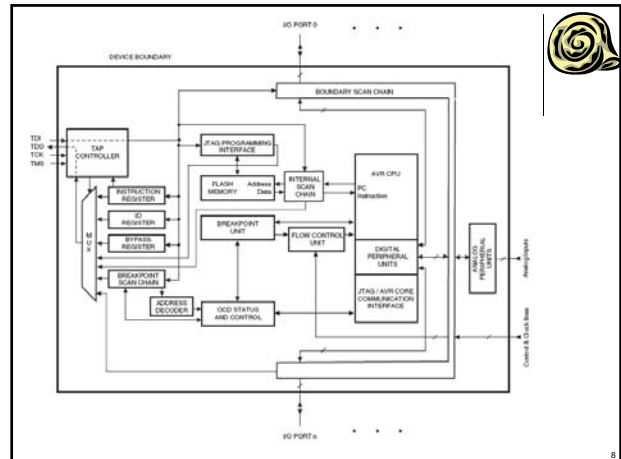
## TAP – Test Access Port

- TMS – test mode select
- TC – test clock
- TDI – test data in
- TDO – test data out  
(TRST signal is not provided)



- JTAGEN fuse programmed
- JDT bit in MCUCSR cleared

7



8

## MCU Control and Status Register

MCUCSR <sub>7</sub>	6	5	4	3	2	1	0
JTD	-	-	JTRF	WDRF	BORF	EXTRF	PORF
R/W	R	R	R/W	R/W	R/W	R/W	R/W
0	0	0					

- JTD JTAG Interface Disable
- JTRF JTAG Reset Flag

## AVR JTAG Boundary-scan

- 205 bits of signal values
  - ports A-G, TWI, clock, reset, ADC, comparator

10

## AVR JTAG Programming

- AVR\_RESET
- PROG\_ENABLE
- PROG\_COMMANDS
- PROG\_PAGELOAD
- PROG\_PAGEREAD

11

## JTAG Commands

- |                                  |                                  |                                      |
|----------------------------------|----------------------------------|--------------------------------------|
| 1a. Chip erase                   | 4e. Latch Data                   | 7b. Load Data Byte                   |
| 1b. Poll for chip erase complete | 4f. Write EEPROM Page            | 7c. Write Lock bits                  |
| 2a. Enter Flash Write            | 4g. Poll for Page Write complete | 7d. Poll for Lock bit Write complete |
| 2b. Load Address High Byte       | 5a. Enter EEPROM Read            | 8a. Enter Fuse/Lock bit Read         |
| 2c. Load Address Low Byte        | 5b. Load Address High Byte       | 8b. Read Extended Fuse Byte          |
| 2d. Load Data Low Byte           | 5c. Load Address Low Byte        | 8c. Read Fuse High Byte              |
| 2e. Load Data High Byte          | 5d. Read Data Byte               | 8d. Read Fuse Low Byte               |
| 2f. Latch Data                   | 6a. Enter Fuse Write             | 8e. Read Lock bits                   |
| 2g. Write Flash Page             | 6b. Load Data Low Byte           | 8f. Read Fuses and Lock bits         |
| 2h. Poll for Page Write complete | 6c. Write Fuse Extended Byte     | 9a. Enter Signature Byte Read        |
| 3a. Enter Flash Read             | 6d. Poll for Fuse Write complete | 9b. Load Address Byte                |
| 3b. Load Address High Byte       | 6e. Load Data Low Byte           | 9c. Read Signature Byte              |
| 3c. Load Address Low Byte        | 6f. Write Fuse High Byte         | 10a. Enter Calibration Byte Read     |
| 3d. Read Data Low and High Byte  | 6g. Poll for Fuse Write complete | 10b. Load Address Byte               |
| 4a. Enter EEPROM Write           | 6h. Load Data Low Byte           | 10c. Read Calibration Byte           |
| 4b. Load Address High Byte       | 6i. Write Fuse Low Byte          | 11a. Load No Operation Command       |
| 4c. Load Address Low Byte        | 6j. Poll for Fuse Write complete |                                      |
| 4d. Load Data Byte               | 7a. Enter Lock bit Write         |                                      |

12

## Programming



Before programming the Flash a Chip Erase must be performed.

1. Enter JTAG instruction PROG\_COMMANDS.
2. Enable Flash write using programming instruction 2a.
3. Load address high byte using programming instruction 2b.
4. Load address low byte using programming instruction 2c.
5. Load data using programming instructions 2d, 2e and 2f.
6. Repeat steps 4 and 5 for all instruction words in the page.
7. Write the page using programming instruction 2g.
8. Poll for Flash write complete using programming instruction 2h, or wait for tWLRH
9. Repeat steps 3 to 7 until all data have been programmed.

13

## Programming via PROG\_PAGELOAD



1. Enter JTAG instruction PROG\_COMMANDS.
2. Enable Flash write using programming instruction 2a.
3. Load the page address using programming instructions 2b and 2c. PCWORD is used to address within one page and must be written as 0.
4. Enter JTAG instruction PROG\_PAGELOAD.
5. Load the entire page by shifting in all instruction words in the page, starting with the LSB of the first instruction in the page and ending with the MSB of the last instruction in the page.
6. Enter JTAG instruction PROG\_COMMANDS.
7. Write the page using programming instruction 2g.
8. Poll for Flash write complete using programming instruction 2h, or wait for tWLRH
9. Repeat steps 3 to 8 until all data have been programmed.

14

## AVR JTAG On-chip debugging



- 4 private JTAG instructions ☹

“The On-chip debug support is considered being private JTAG instructions, and distributed within ATMEL and to selected third-party vendors only”

- **PRIVATE0; \$8** Private JTAG instruction for accessing On-chip Debug system.
- **PRIVATE1; \$9** Private JTAG instruction for accessing On-chip Debug system.
- **PRIVATE2; \$A** Private JTAG instruction for accessing On-chip Debug system.
- **PRIVATE3; \$B** Private JTAG instruction for accessing On-chip Debug system.

15

## Breakpoint unit



- change of program flow
- single step break
- 2 program memory breakpoints
- 2 combined breakpoints (program/data/range)

16