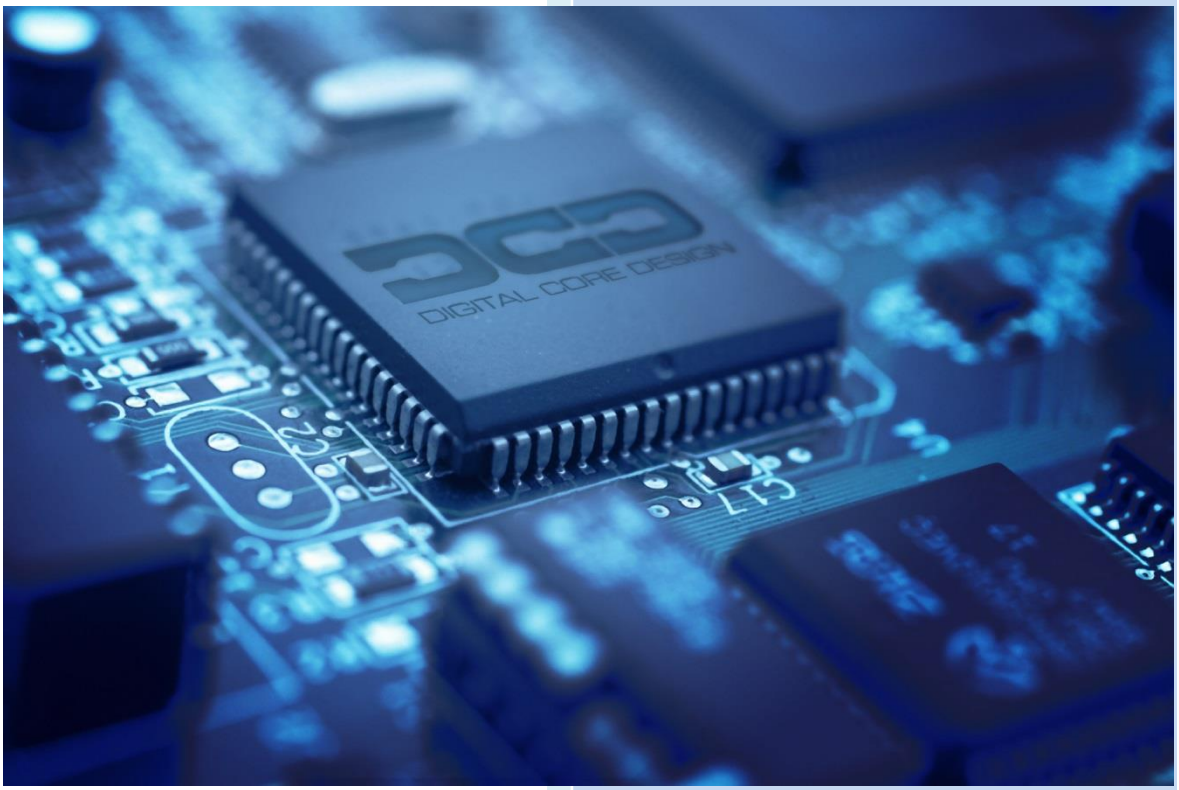
DIGITAL CORE DESIGN

2018

DoCD IP Core



DCD on Chip Debug System v. 6.02

COMPANY OVERVIEW

Digital Core Design is a leading IP Core provider and a System-on-Chip design house. The company was founded in 1999 and since the very beginning has been focused on IP Core architecture improvements. Our innovative, silicon proven solutions have been employed by over 300 customers and with more than 500 hundred licenses sold to companies like Intel, Siemens, Philips, General Electric, Sony and Toyota. Based on more than 70 different architectures, starting from serial interfaces to advanced microcontrollers and SoCs, we are designing solutions tailored to your needs.

IP CORE OVERVIEW

DCD's on-chip Debug System (DoCD™) prominently cuts debugging time. Integrating DCD IP Cores with a Hardware Assisted Debugger and Debug IP Core, provides a powerful SoC development tool, with advanced features. The DoCD™ system consists of three major blocks:

- Debug IP Core
- Hardware Assisted Debugger
- Debug Software

The Debug IP Core block is a **real-time hardware debugger**, which provides an access to all chip registers, memories and peripherals, connected to DCD's IP Core (Dx8051/Dx80390 /DRPIC/ DFPIC/Dx6811) and controls CPU work by **non-intrusive** method. A high-performance Hardware Assisted Debugger (**USB-xTAG**) is connected to the target system containing the DCD's core either in FPGA or ASIC. The Hardware Assisted Debugger manages communication between the Debug IP Core inside silicon using DTAG, TTAG, or JTAG protocol, and Debug Software using USB port.

FLASH PROGRAMMING

The DoCD debugger fully supports programming of all FLASH memory devices. Such support is assured by configurability of FLASH programming algorithm and supported devices' database. New FLASH device can be easily added to the existing base, using a built-in editor. The DoCD debugger allows the user to easily perform in-system programming of its FLASH memory, without using any external equipment. The FLASH programming task is performed directly within the Debug software and after uploading of the code, it is ready for the debugging. Due to HAD2 support, the programming time is very short. This feature saves time and makes usage of DCD's debugger very comfortable and flexible.

NON-INTRUSIVE SYSTEM

In typical intrusive systems, the debugging tool uses for its own needs some system resources, e.g.: part of program space, several cells of RAM memory, ports' pins; sometimes system is losing interrupts or the program code is manipulated, to support software breakpoints and so on. Even a simple debugging system consumes UART and timer resources, to support its own tasks. These simple 'emulators' cannot provide trace and other advanced debugging functions, being at the same time very intrusive in the debugging cycle. Imagine, trying to debug an interrupt problem, while the 'emulator' is manipulating interrupts itself! Developing firmware is all about producing code, which is 100% reliable in operation and fully understood, in terms of how it will perform in adverse conditions. **A real-time, non-intrusive on-chip debugger**, which assists the user in his tasks, is the most important tool that the user can have. This is why using of non-intrusive systems is so important. It is also the reason, why the **DoCD™ debugging tool has been designed as a non-intrusive system**.

KEY FEATURES OF THE DEBUG IP CORE

- Processor execution control
  - *Run, Halt*
  - *Reset*
  - *Step into instruction*
  - *Skip Instruction*
- Read-write all processor contents
  - *Program Counter (PC)*
  - *Program Memory*
  - *Internal (direct) Data Memory*
  - *Special Function Registers (SFRs)*
  - *External Data Memory*
- Code execution breakpoints
  - ***up to eight** real-time PC breakpoints*
  - *unlimited number of real-time OPCODE breakpoints (v 4.00 and above)*
- Hardware execution watch-points
  - *two at Internal (direct) Data Memory*
  - *two at Special Function Registers (SFRs)*
  - *two at External Data Memory*
- Hardware watch-points activated at:
  - *certain address by any write into memory*
  - *certain address by any read from memory*
  - *certain address by write into memory a required data*
  - *certain address by read from memory a required data*
- An unlimited number of software watch-points
  - *Internal (direct) Data Memory*
  - *Special Function Registers (SFRs)*
  - *External Data Memory*
- An unlimited number of software breakpoints
  - *Program Memory(PC)*
- Instructions Smart Trace Buffer – configurable up to 8192 levels (optional)

- Automatic adjustment of debug data transfer speed rate between HAD and Silicon
- Communication interface
  - *TTAG interface – v4.70 and above*
  - *JTAG interface – v4.00 and above*
  - *DTAG three wire communication – v3.xx*
- Fully static synchronous design with no internal tri-states

DEBUG SOFTWARE

The **DoCD™** Software (DS) is a Windows based application. It is fully compatible with all existing 8051/80251/80390 C compilers and Assemblers. The DS was designed to work in two major modes: a software simulator mode and a hardware debugger mode. Those two modes allow the pre-silicon software validation in simulation mode; then real-time debugging of the developed software inside silicon - using the debugger mode. Once loaded, the program may be observed in the Source Window, run at full-speed, single stepped by the machine or C-level instructions or stopped at any of the breakpoints and watch-points. The DoCD™ Debug Software supports all DCD's DQ8051x/DT8051x /DP8051x /DP80390x in the following architectures:

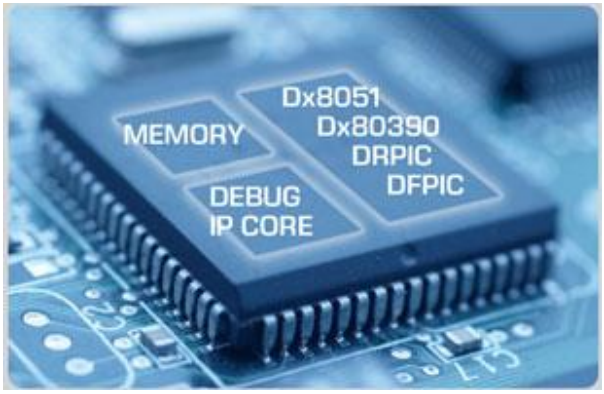
- Revolutionary Quad-Pipelined Ultra High Performance (DQ - 1 CLK per instruction)
- Pipelined High Performance– (DP - 1T cycle)
- Tiny RISC – (DT -2T cycle) with their particular configurations.

REAL-TIME HARDWARE DEBUGGER

The real-time hardware debugger - a tool able to detect processor's internal properties, which are not visible outside the processor without any violation of real-time operations. The DoCD™ gives you the chance to track down hidden bugs within the application running with a microcontroller. Internal events, such as the reading of the SFR's-registers, are not mirrored on the external address-data bus. However, by using special logic to detect operations that affect internal resources, the DoCD™ gives you the ability to track such internal events, without any violation of a real-time operation. There is no need to use any special external logic for the emulation.

DEBUG IP CORE

The Debug IP Core can be provided as a VHDL or Verilog source code, as well as CPLD/FPGA EDIF Netlist, depending on customer's requirements. Due to the fact, that many SoC designs have both power and area limitations, the DoCD™ provides a scaled solution. The Debug IP Core can be scaled to control gate count - the benefit is fewer gates - for lower use of power and core size, while maintaining excellent debug abilities. Typically, all the features are utilized in pre-silicon debug (i.e. hardware emulation or FPGA evaluation) with less features availed in the final silicon.



HAD2 UNIT KEY FEATURES

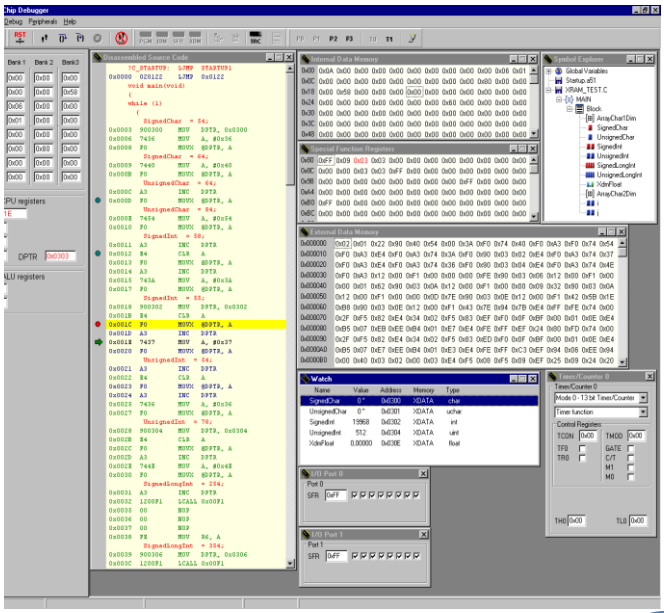
- USB communication interface to target host at FULL speed
- Synchronous communication interface to Debug IP Core
  - *TTAG interface – Debug IP v4.70*
  - *JTAG interface – Debug IP v4.xx, v6.xx*
  - *DTAG interface – Debug IP v3.xx*
- Supports following I/O voltage standards
  - *3.3 Volt systems*
  - *2.5 Volt systems*
  - *1.8 Volt systems*
- Single power supply directly from USB host
- Small physical dimensions – pendrive package

HARDWARE ASSISTED DEBUGGER

The Hardware Assisted Debugger (HAD) is a small hardware adapter, that manages communication between the Debug IP Core inside silicon and a USB port of the host PC, running the DoCD™ Debug Software.

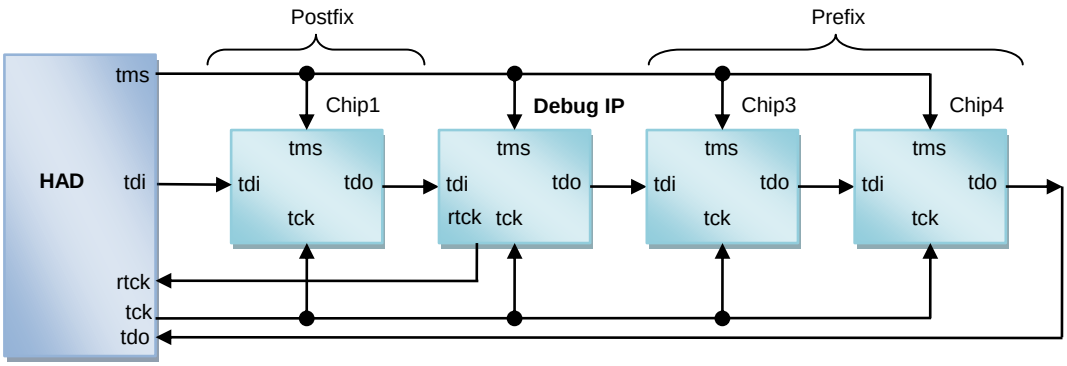
DEBUG SOFTWARE KEY FEATURES

- In-system FLASH programming
- Three working modes
  - *hardware debugger*
  - *in circuit emulator*
  - *software simulator*
- Source Level Debugging:
  - *C level hardware/software breakpoints*
  - *C code execution*
    - *line by line*
    - *over line*
    - *out of function*
    - *skip line*
  - *ASM code execution*
    - *Instruction by instruction*
    - *over instruction*
    - *out of function*
    - *skip instruction*
  - *ASM, C source view of code*
- Symbol Explorer provides hierarchical tree view of all symbols:
  - *modules*
  - *functions*
  - *blocks*
  - *variables and more*
- Contents sensitive Watch window:
  - *Local variables view*
  - *Up to 3 independent watches*
- Symbolic debug including:
  - *code*
  - *variables*
  - *variable types*
- **Unlimited** number of real-time hardware break-points
  - *Program Memory (CODE)*
- Two real-time hardware watch-points for each space:
  - *Internal (direct) Data Memory (IDM)*
  - *Special Function Registers (SFR)*
  - *eXternal Data Memory (XDM)*
- Unlimited number of software breakpoints
  - *Program Memory*
  - *Internal (direct) Data Memory (IDM)*
  - *Special Function Registers (SFR)*
  - *eXternal Data Memory (XDM)*
- Set/clear software or hardware breakpoints, watch-points in Disassembled and C Source Code windows
- 1024 steps deep Software Trace
- Load Program Memory content from:
  - *OMF-51, extended OMF-51 files*
  - *OMF-251 file*
  - *CDB object format*
  - *Intel HEX-51, HEX-386 files*
  - *BIN file*
- Auto refresh of all windows during execution of program
  - *Registers' panel including ACC, B, PSW, PC, SP, DPTR, DPP and four banks of general purpose registers R0-R7*
  - *Internal (direct) Data Memory (IDM)*
  - *Special Function Registers (SFR)*
  - *eXternal Data Memory (XDM)*
  - *Timers/Counters*
  - *UARTs*
  - *I/O Ports*
- Dedicated windows for peripherals
- Configurable auto refresh time period with 1s step resolution
- Status bar containing number of actually executed instructions, number of clock periods and real processor speed rate
- Hardware Assisted Debugger interface
  - *TTAG interface*
  - *JTAG interface*
  - *DTAG interface*
- The system runs on Windows® 2000/2003/XP/7/8/8.1 (both 32 and 64 bit) PC
- Supports software tools from Keil, IAR, SDCC and others



DOCD IN JTAG CHAIN

The DoCD™ debug IP Core v 4.00 (and higher) can be used as a standalone device or can be plugged into a JTAG chain. It means, that standard JTAG pins can be used and other JTAG devices can be controlled, along with the DoCD Debug IP. Such solution saves off-chip pins of ASIC/FPGA device.

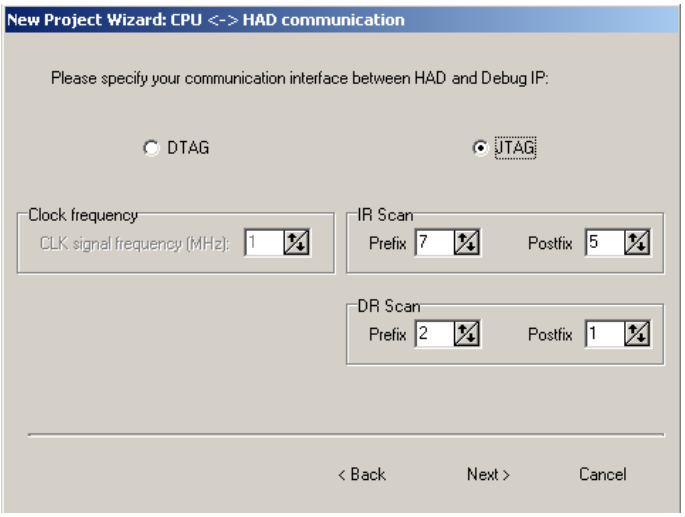


The example target shown in the figure above consists of the DoCD Debug IP and three, fully JTAG compliant devices. The Chip1 has 5-bit long IR (Instruction Register), Chip3 IR's has 3-bit long and Chip4 has 4-bit long IR. A DR (Data Register) is always 1-bit long, for each JTAG device.

The following values should be written into DoCD Windows Debug Software configuration window:

- IR–prefix (3+4), DR-prefix (1+1)
- IR–postfix (5), DR-postfix (1)

It is shown in the following figure.



The 0 value should be written in an appropriate IR and DR field, in case where there wouldn't be any prefix or postfix devices.

DEBUG IP CORE PINOUT

The following pins are used by DoCD™ debug IP Core.

A. TTAG interface – v 4.70 and above:

PIN	TYPE	DESCRIPTION	
ttdi	input	DoCD™ data input	Connected together as **ttdio** pin
ttdoen	output	DoCD™ data output enable	
ttdo	output	DoCD™ data output	
ttck	output	DoCD™ clock line	

B. JTAG interface – v 4.00 and above:

PIN	TYPE	DESCRIPTION
tdi	input	DoCD™ TAP data input
tck	input	DoCD™ TAP clock line
tms	input	DoCD™ TAP mode select
tdo	output	DoCD™ TAP data output
rtck	output	DoCD™ return clock

C. DTAG interface – v 3.xx :

PIN	TYPE	DESCRIPTION
docddatai	input	DoCD™ data input
docddatao	output	DoCD™ data output
docdclk	output	DoCD™ clock line

AREA UTILIZATION

The following table gives a survey about the Debug IP Core area, in the FPGA and ASIC devices.

Device vendor		Area DTAG	Area JTAG	Area TTAG
ALTERA	[LCs]	720	600	470
XILINX	[Slices]	360	300	240
LATTICE	[LUT4s]	720	610	500
ASIC	[gates]	2500	2100	1650

SYSTEM FEATURES

- FLASH PROGRAMMING:

All FLASH memory devices are supported by the DoCD debug system. Such support is assured by configurability of a FLASH programming algorithm and devices' database. A new FLASH device can be easily added to the existing database, by using a built-in editor. The DoCD debugger allows simple in-system programming of its FLASH memory, without using any additional tools. The DoCD programming task is performed directly within the Debug software and after uploading of the core, it is ready for debugging. The programming time is very short and write operations are performed by certain FLASH device, with maximum speed allowed.

- HARDWARE BREAKPOINTS:

The number of hardware breakpoints is **unlimited**. Like software breakpoints, hardware execution breakpoints can be set in the Program Memory space. They stop program execution just **prior** an instruction is pointed by the Program Counter (PC). In other words, instruction located at the PC breakpoint address, is not executed. The difference is in the method of program execution. In this case, program is run with full clock speed (in real-time) and the processor is halted, when the hardware signalizes a real breakpoint condition.

- HARDWARE WATCH-POINTS:

The number of hardware watch-points is limited to six in different address spaces. Like software breakpoints, hardware execution watch-points can be set in direct RAM, SFRs and external RAM. They stop program execution after an instruction is being executed. The difference is in the method of program execution. In this case, the program is run with full clock speed (in real-time) and the processor is halted, when the hardware signalizes a real watch-point condition.

- SOFTWARE BREAKPOINTS:

An unlimited number of software breakpoints can be set anywhere in the physical address space of the processor (in Program Memory space, direct RAM, SFRs and external RAM). If at least one software breakpoint is set, the program is executed in an automatic step by step mode, with checking if a certain breakpoint condition is met. Program execution is halted, when the breakpoint condition is already met and its execution can be resumed at any time, in any appropriate mode.

- MIXED MODE BREAKPOINTS:

The mixed breakpoint mode is also available and it means, that software and hardware breakpoints and watch-points are mixed in the system. This gives you the flexibility in the debugging - for example, two different break conditions can be set, by using watch-points and hardware breakpoints. In each breakpoint mode halt means: CPU is halted and instructions are no longer being fetched and all internal peripherals are also stopped (e.g. timers, watchdog). The UARTs work correctly in any case.

- SYMBOL EXPLORER:

The Symbol Explorer provides a hierarchical tree view of all C project symbols. It supports all C types, variables, constants, functions and symbolic names of registers. Along with a watch window, it provides a flexible and powerful debugging feature at high C language level.

- SCALED SOLUTION:

Due to the fact, that many SoC designs have both power and gate limitations, DCD provides a scaled solution. Debug extensions can be scaled to control gate counts. The benefit is fewer gates, for lower use of power and core size, while maintaining excellent debug abilities.

- HOST REQUIREMENTS:

A Pentium class computer (with minimum 512 MB of memory, 32 MB of free space on Hard Disk, CD-ROM drive, USB port and Windows 2000/2003/XP/7 operating system) is required.

CONTACT

Digital Core Design Headquarters:

Wroclawska 94, 41-902 Bytom, POLAND

e-mail: info@dcd.pl
tel.: 0048 32 282 82 66
fax: 0048 32 282 74 37

Distributors:

Please check: http://dcd.pl/sales