DP8051 IP Core

2017

Pipelined High Performance 8-bit Microcontroller v.4.01
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

The DP8051 is an ultra-high performance, speed optimized soft core of a single-chip 8-bit embedded controller, dedicated to operate with fast (typically on-chip) and slow (off-chip) memories. The core has been designed with a special concern for performance to power consumption ratio. This ratio is extended by an advanced power management unit - the PMU. The DP8051 soft core is 100% binary-compatible with the industry standard 8051 8-bit microcontroller. There are two configurations of the DP8051: Harvard, where an internal data and program buses are separated and von Neumann, with common program and external data bus. The DP8051 has a Pipelined RISC architecture and executes 120-300 million instructions per second. Dhrystone 2.1 benchmark program runs from 11.46 to 15.55 times faster than the original 80C51 at the same frequency. This performance can be also exploited to a great advantage in low power applications, where the core can be clocked over ten times slower than the original implementation, with no performance penalty. The DP8051 is delivered with fully automated test bench and complete set of tests, allowing easy package validation at each stage of SoC design flow.

LICENSING

Comprehensible and clearly defined licensing methods without royalty-per-chip fees make use of our IP Cores easy and simple.

Single-Site license option – dedicated to small and middle sized companies, which run their business in one place.

Multi-Site license option – dedicated to corporate customers, who operate at several locations. The licensed product can be used in selected company branches.

In all cases the number of IP Core instantiations within a project and the number of manufactured chips are unlimited. The license is royalty-per-chip free. There are no restrictions regarding the time of use.

There are two formats of the delivered IP Core:

> VHDL or Verilog RTL synthesizable HDL Source code
> FPGA EDIF/NGO/NGD/QXP/VQM Netlist

CPU FEATURES

- 100% software compatible with 8051 industry standard
- Pipelined RISC architecture enables to execute 15.55 times faster than the original 80C51 at the same frequency
- Up to 14.632 VAX MIPS at 100 MHz
- 24 times faster multiplication
- 12 times faster addition
- Up to 256 bytes of internal (on-chip) Data Memory
- Up to 64K bytes of internal (on-chip) or external (off-chip) Program Memory
- Up to 16M bytes of external (off-chip) Data Memory
- User programmable Program Memory Wait States solution, for wide range of memories speed
- User programmable External Data Memory Wait States solution, for wide range of memories speed
- De-multiplexed Address/Data bus to allow easy connection to memory
- Dedicated signal for Program Memory writes.
- Interface for additional Special Function Registers
- Fully synthesizable, static synchronous design, with positive edge clocking and no internal tri-states
- Scan test ready

SYMBOL

- port0i
- port1i
- port2i
- port3i
- ramdatai
- sfrdatai
- prgromdatai
- prgramdatai
- xdatai
- xdataw
- xdatao
- xdataz
- xdatard
- xdatawr
- xprgrd
- xprgwr
- int0
- int1
- ramdata
- ramaddr
- ramdword
- raman
- rameo
- sfraddr
- sfrdatao
- sfrwe
- stop
- pmm
- rxdoi
- bndo
- cdon
- debugac
- rsto
C O N F I G U R A T I O N

The following parameters of the DP8051 core can be easily adjusted to requirements of a dedicated application and technology. The configuration of the core can be effortlessly done, by changing appropriate constants in the package file. There is no need to change any parts of the code.

- Internal Program Memory type
- Internal Program ROM Memory size
- Internal Program RAM Memory size
- Internal Program Memory fixed size
- Interrupts
- Power Management Mode
- Stop mode
- DoCD™ debug unit

Except parameters mentioned above, all available peripherals and external interrupts can be excluded from the core, by changing appropriate constants in the package file.

P E R I P H E R A L S

- DoCD™ debug unit
  - Processor execution control
    - Run, Halt
    - 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 OP/PCODE breakpoints
  - Hardware execution watch-points at
    - Internal (direct) Data Memory
    - Special Function Registers (SFRs)
    - External Data Memory
  - Hardware watch-points activated at a certain
    - address by any write into memory
    - address by any read from memory
    - address by write required data into memory
    - address by read required data from memory
  - Instructions Smart Trace Buffer – configurable up to 8192 levels (optional)
  - Automatic adjustment of debug data transfer speed rate between HAD and Silicon
  - TTAG or JTAG Communication interface

- Power Management Unit
  - Power management mode
  - Switchback feature
  - Stop mode

- Interrupt Controller
  - 2 priority levels
  - 2 external interrupt sources
  - 3 interrupt sources from peripherals

- Four 8-bit I/O Ports
  - Bit addressable data direction for each line
  - Read/write of single line and 8-bit group

- Two 16-bit timer/counters
  - Timers clocked by internal source
  - Auto reload 8-bit timers

- Extensively gated event counters
- Full-duplex serial port
- Synchronous mode, fixed baud rate
- 8-bit asynchronous mode, fixed baud rate
- 9-bit asynchronous mode, fixed baud rate
- 9-bit asynchronous mode, variable baud rate

P I N S  D E S C R I P T I O N

<table>
<thead>
<tr>
<th>PIN</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>input</td>
<td>Global clock</td>
</tr>
<tr>
<td>reset</td>
<td>input</td>
<td>Global reset input</td>
</tr>
<tr>
<td>port0i</td>
<td>input</td>
<td>Port 0 input</td>
</tr>
<tr>
<td>port1i</td>
<td>input</td>
<td>Port 1 input</td>
</tr>
<tr>
<td>port2i</td>
<td>input</td>
<td>Port 2 input</td>
</tr>
<tr>
<td>port3i</td>
<td>input</td>
<td>Port 3 input</td>
</tr>
<tr>
<td>iprogramsize</td>
<td>input</td>
<td>Size of on-chip RAM CODE</td>
</tr>
<tr>
<td>iprogramsize</td>
<td>input</td>
<td>Size of on-chip ROM CODE</td>
</tr>
<tr>
<td>programdata</td>
<td>input</td>
<td>Data bus from int. RAM prog. memory</td>
</tr>
<tr>
<td>programdata</td>
<td>input</td>
<td>Data bus from int. ROM prog. memory</td>
</tr>
<tr>
<td>sxmdatini</td>
<td>input</td>
<td>Data bus from sync external data memory (SXDM)</td>
</tr>
<tr>
<td>xdatai</td>
<td>input</td>
<td>Data bus from external memories</td>
</tr>
<tr>
<td>ready</td>
<td>input</td>
<td>External memory data ready</td>
</tr>
<tr>
<td>ramdatini</td>
<td>input</td>
<td>Data bus from internal data memory</td>
</tr>
<tr>
<td>sfrdatini</td>
<td>input</td>
<td>Data bus from user SFR's</td>
</tr>
<tr>
<td>init</td>
<td>input</td>
<td>External interrupt 0</td>
</tr>
<tr>
<td>int1</td>
<td>input</td>
<td>Internal interrupt 1</td>
</tr>
<tr>
<td>t0</td>
<td>input</td>
<td>Timer 0 output</td>
</tr>
<tr>
<td>t1</td>
<td>input</td>
<td>Timer 1 output</td>
</tr>
<tr>
<td>gate0</td>
<td>input</td>
<td>Timer 0 gate input</td>
</tr>
<tr>
<td>gate1</td>
<td>input</td>
<td>Timer 1 gate input</td>
</tr>
<tr>
<td>rxdio</td>
<td>input</td>
<td>Serial receiver input 0</td>
</tr>
<tr>
<td>tdi</td>
<td>input</td>
<td>DoCD™ TAP data input</td>
</tr>
<tr>
<td>tck</td>
<td>input</td>
<td>DoCD™ TAP clock input</td>
</tr>
<tr>
<td>tms</td>
<td>input</td>
<td>DoCD™ TAP mode select input</td>
</tr>
<tr>
<td>rsto</td>
<td>output</td>
<td>Reset output</td>
</tr>
<tr>
<td>port0o</td>
<td>output</td>
<td>Port 0 output</td>
</tr>
<tr>
<td>port1o</td>
<td>output</td>
<td>Port 1 output</td>
</tr>
<tr>
<td>port2o</td>
<td>output</td>
<td>Port 2 output</td>
</tr>
<tr>
<td>port3o</td>
<td>output</td>
<td>Port 3 output</td>
</tr>
<tr>
<td>prgaddr</td>
<td>output</td>
<td>Internal program memory address bus</td>
</tr>
<tr>
<td>prgdatato</td>
<td>output</td>
<td>Data bus for internal program memory</td>
</tr>
<tr>
<td>prgdatawi</td>
<td>output</td>
<td>Internal program memory write</td>
</tr>
<tr>
<td>sxmdatado</td>
<td>output</td>
<td>Sync XDATA memory address bus (SXDM)</td>
</tr>
<tr>
<td>sxmdatato</td>
<td>output</td>
<td>Data bus for Sync XDATA memory (SXDM)</td>
</tr>
<tr>
<td>sxmdoe</td>
<td>output</td>
<td>Sync XDATA memory read (SXDM)</td>
</tr>
<tr>
<td>sxmdwe</td>
<td>output</td>
<td>Sync XDATA memory write (SXDM)</td>
</tr>
<tr>
<td>xaddr</td>
<td>output</td>
<td>Address bus for external memories</td>
</tr>
<tr>
<td>xdatato</td>
<td>output</td>
<td>Data bus for external memories</td>
</tr>
<tr>
<td>xdata</td>
<td>output</td>
<td>Turn xdata bus into ‘2’ state</td>
</tr>
<tr>
<td>xprgrd</td>
<td>output</td>
<td>External program memory read</td>
</tr>
<tr>
<td>xprogwr</td>
<td>output</td>
<td>External program memory write</td>
</tr>
<tr>
<td>xdatard</td>
<td>output</td>
<td>External data memory read</td>
</tr>
<tr>
<td>xdataawr</td>
<td>output</td>
<td>External data memory write</td>
</tr>
<tr>
<td>ramaddr</td>
<td>output</td>
<td>Internal Data Memory address bus</td>
</tr>
<tr>
<td>ramdatato</td>
<td>output</td>
<td>Data bus for internal data memory</td>
</tr>
<tr>
<td>ramoe</td>
<td>output</td>
<td>Internal data memory output enable</td>
</tr>
<tr>
<td>ramwe</td>
<td>output</td>
<td>Internal data memory write enable</td>
</tr>
<tr>
<td>sfraddr</td>
<td>output</td>
<td>Address bus for user SFR’s</td>
</tr>
<tr>
<td>sfrdatato</td>
<td>output</td>
<td>Data bus for user SFR’s</td>
</tr>
<tr>
<td>sfrwe</td>
<td>output</td>
<td>User SFR’s writable enable</td>
</tr>
<tr>
<td>sfrwe</td>
<td>output</td>
<td>User SFR’s write enable</td>
</tr>
<tr>
<td>tdo</td>
<td>output</td>
<td>DoCD™ TAP data output</td>
</tr>
<tr>
<td>rtk</td>
<td>output</td>
<td>DoCD™ return clock line</td>
</tr>
<tr>
<td>debugacs</td>
<td>output</td>
<td>DoCD™ accessing data</td>
</tr>
<tr>
<td>coderun</td>
<td>output</td>
<td>CPU is executing an instruction</td>
</tr>
<tr>
<td>pmm</td>
<td>output</td>
<td>Power management mode indicator</td>
</tr>
<tr>
<td>stop</td>
<td>output</td>
<td>Stop mode indicator</td>
</tr>
<tr>
<td>rxdio</td>
<td>output</td>
<td>Serial receiver output 0</td>
</tr>
<tr>
<td>txdio</td>
<td>output</td>
<td>Serial transmitter output 0</td>
</tr>
</tbody>
</table>

Copyright © 1999-2017 DCD – Digital Core Design. All Rights Reserved. All trademarks mentioned in this document are the property of their respective owners.
The DP8051 soft core is dedicated for operation with wide range of Program and Data memories. Slow Program and External Data memory may assert a memory WAIT signal to hold up CPU activity, for required period of time.

**Program Code Space Implementation**

The following figure shows an example Program Memory space implementation in systems with the DP8051 Microcontroller core. The on-chip Program Memory located in address space between 0kB and 1kB, is typically used for BOOT code, with system initialization functions. This part of the code is typically implemented as ROM. The on-chip Program Memory located in address space between 60kB and 64kB, is typically used for timing critical part of the code e.g. interrupt subroutines, arithmetic functions etc. This part of the code is typically implemented as RAM and can be loaded by the BOOT code, during initialization phase from an off-chip memory or through a KS232 interface from an external device. From the two spaces mentioned above, the program code is executed without wait-states and can achieve top performance, up to 200 million instructions per second (many instructions executed in one clock cycle). The off-chip Program Memory located in address space between 1kB and 60kB, is typically used for a main code and constants. This part of the code is usually implemented as ROM, SRAM or FLASH device. Due to relatively long access time, the program code executed from devices mentioned earlier must be fetched with additional Wait-States. The number of required Wait-States depends on a memory access time and the DP8051 clock frequency. In most cases, the proper number of Wait-States cycles is in range 2 to 5. The READY pin can be also dynamically modulated e.g. by SDRAM controller. The following figure shows typical Program Memories connection in system with the DP8051 Microcontroller core.

**Design Features**

- **Program Memory:**
  - The DP8051 soft core is dedicated to operate with Internal and External Program Memory. Internal Program Memory can be implemented as:
    - ROM located in address range between 0x0000 to (RAMmax-1)
    - RAM located in address range between [RAMmax-1] to 0xFFFF
  - External Program Memory can be implemented as ROM or RAM located in address range between ROMmax to RAMmax.

- **Internal Data Memory:**
  - The DP8051 can address Internal Data Memory of up to 256 bytes. The Internal Data Memory can be implemented as Single-Port synchronous RAM.

- **External Data Memory:**
  - The DP8051 soft core can address up to 16 MB of External Data Memory. Extra DPX (Data Pointer eXtended) register is used for segments swapping.

- **User Special Function Registers:**
  - Up to 104 External (user) Special Function Registers (ESFRs) may be added to the DP8051 design. ESFRs are memory mapped into Direct Memory between addresses 0x80 and 0xFF in the same manner, as core SFRs and may occupy any address that is not occupied by a SFR core.

- **Wait States Support:**
All Program Memory spaces are fully configurable. For timing-critical applications the whole program code can be implemented as on-chip ROM and (or) RAM and executed without Wait-States, but for some other applications, the whole program code can be implemented as off-chip ROM or FLASH and executed with required number Wait-State cycles.

**UNITS SUMMARY**

**ALU** – Arithmetic Logic Unit - performs the arithmetic and logic operations, during execution of an instruction. It contains accumulator (ACC), Program Status Word (PSW), (B) registers and related logic, like arithmetic unit, logic unit, multiplier and divider.

**Opcode Decoder** – Performs an opcode decoding instruction and control functions for all other blocks.

**Control Unit** – It performs the core synchronization and data flow control. This module is directly connected to Opcode Decoder and it manages execution of all microcontroller tasks.

**Program Memory Interface** – Program Memory Interface contains Program Counter (PC) and related logic. It performs the instructions code fetching. Program Memory can be also written. This feature allows usage of a small boot loader, to load new program into ROM, RAM, EPROM or FLASH EEPROM storage via UART, SPI, I2C or DoCD™ module.

**External Memory Interface** - Contains memory access related registers, such as Data Page High (DPH), Data Page Low (DPL) and Data Page Pointer (DPP) registers. It performs the external Program and Data Memory addressing and data transfers. Program fetch cycle length can be programmed by the user. This feature is called Program Memory Wait States and it allows core, to work with different speed program memories.

**Synchronous eXternal Data Memory (SXDM) Interface** – contains XDATA memory access related logic, allowing fast access to synchronous memory devices. It performs the external Data Memory addressing and data transfers. This memory can be used to store large variables, frequently accessed by CPU, improving overall performance of application.

**Internal Data Memory Interface** – Interface controls access into the internal memory of size up to 256 bytes. It contains 8-bit Stack Pointer (SP) register and related logic.

**User SFRs Interface** – Special Function Registers interface controls access to the special registers. It contains standard and used defined registers and related logic. User defined external devices can be quickly accessed (read, written, modified), by using all direct addressing mode instructions.

**Interrupt Controller** – Interrupt Controller module is responsible for the interrupt manage system of the external and internal interrupt sources. It contains interrupt related registers, such as Interrupt Enable (IE), Interrupt Priority (IP) and (TCON) registers.

**Timers** – System timers module. Contains two 16 bits configurable timers: Timer 0 (TH0, TL0), Timer 1 (TH1, TL1) and Timers Mode (TMOD) registers. In the timer mode, timer registers are incremented every 12 (or 4) CLK periods, when appropriate timer is enabled. In the counter mode, the timer registers are incremented every falling transition on their corresponding input pins (T0, T1), if gates are opened (GATE0, GATE1). T0, T1 input pins are sampled every CLK period. It can be used as clock source for UARTs.

**UART0** – Universal Asynchronous Receiver and Transmitter module is full duplex, which means, it can transmit and receive concurrently. Includes Serial Configuration register (SCON), serial receiver and transmitter buffer (SBUF) registers. Its receiver is double-buffered, meaning, it can commence reception of a second byte, before the previously received byte has been read from the receive register. Writing to SBUFO loads the transmit register and reading SBUFO, reads a physically separate receive register. Works in 3 asynchronous and 1 synchronous modes. UART0 can be synchronized by Timer 1 or Timer 2 (if present in the system).

**Ports** - Block contains 8051’s general purpose I/O ports. Each of port’s pin can be read/write as a single bit or as a 8-bit bus P0, P1, P2, P3.

**Power Management Unit** – Power Management Unit contains advanced power saving mechanisms with switchback feature, allowing external clock control logic to stop clocking (Stop mode) or run core in lower clock frequency (Power Management Mode), to significantly reduce power consumption. Switchback feature allows UARTs and interrupts to be processed in full speed mode, if enabled. It is highly desirable, when microcontroller is planned to be used in portable and power critical applications.

**DoCD™ Debug Unit** – it’s a real-time hardware debugger, which provides debugging capability of a whole SoC system. Unlike other on-chip debuggers, DoCD™ ensures non-intrusive debugging of running application. It can halt, run, step into or skip an instruction, read/write any contents of microcontroller, including all registers, internal and external program memories and all SFRs, including user defined peripherals. Hardware breakpoints can be set and controlled on program memory, internal and external data memories, as well as on SFRs. Hardware watchpoints can be set and controlled on internal and external data memories and also
on SFRs. Hardware watchpoints are executed, if any write/read occurs at particular address, with certain data pattern or without pattern. Two additional pins: CODERUN and DEBUGACS, indicate the state of the debugger and CPU. CODERUN is active, when CPU is executing an instruction. DEBUGACS pin is active, when any access is performed by DoCD™ debugger. The DoCD™ system includes TTAG or JTAG interface and complete set of tools, to communicate and work with core in real time debugging. It is built, as a scalable unit and some features can be turned off by the user, to save silicon and reduce power consumption. When debugger is not used, it is automatically switched to power save mode. Finally, when debug option is no longer used, whole debugger is turned off.

**DELIVERABLES**

- Source code:
  - VHDL Source Code or/and
  - VERILOG Source Code or/and
  - Encrypted, or plain text EDIF
- VHDL & VERILOG test bench environment
  - Active-HDL automatic simulation macros
  - ModelSim automatic simulation macros
  - Tests with reference responses
- Technical documentation
  - Installation notes
  - HDL core specification
  - Datasheet
- Synthesis scripts
- Example application
- Technical support
  - IP Core implementation support
  - 3 months maintenance
  - Delivery of the IP Core and documentation updates, minor and major versions changes
  - Phone & email support

**DP8051 FAMILY OVERVIEW**

Main features of each DP8051 family member have been summarized in the table below. It gives a brief member characteristic, helping you to select the most suitable IP Core for your application. You can specify your own peripheral set (including listed below and others) and requests the core modifications.

<table>
<thead>
<tr>
<th>Design</th>
<th>Program Memory space</th>
<th>Interrupts sources</th>
<th>Debug/Trace</th>
<th>I/O Ports</th>
<th>UART</th>
<th>Timers/Counters</th>
<th>Watchdog</th>
<th>SPI</th>
<th>Fixed Point Coprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP8051CPU</td>
<td>on-chip RAM 64k/64k</td>
<td>on-chip ROM 256</td>
<td>8</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>SPI</td>
<td>-</td>
</tr>
<tr>
<td>DP8051</td>
<td>on-chip RAM 64k/64k</td>
<td>on-chip ROM 256</td>
<td>8</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>SPI</td>
<td>-</td>
</tr>
<tr>
<td>DP8051XP</td>
<td>on-chip RAM 64k/64k</td>
<td>on-chip ROM 256</td>
<td>8</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>SPI</td>
<td>-</td>
</tr>
</tbody>
</table>

**DP80390 FAMILY OVERVIEW**

Main features of each DP80390 family member have been summarized in the table below. It gives a brief member characteristic, helping you to select the most suitable IP Core for your application. You can specify your own peripheral set (including listed below and others) and requests the core modifications.

<table>
<thead>
<tr>
<th>Design</th>
<th>Program Memory space</th>
<th>Interrupts sources</th>
<th>Debug/Trace</th>
<th>I/O Ports</th>
<th>UART</th>
<th>Timers/Counters</th>
<th>Watchdog</th>
<th>SPI</th>
<th>Fixed Point Coprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP80390CPU</td>
<td>on-chip RAM 64k/64k</td>
<td>on-chip ROM 256</td>
<td>8</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>SPI</td>
<td>-</td>
</tr>
<tr>
<td>DP80390</td>
<td>on-chip RAM 64k/64k</td>
<td>on-chip ROM 256</td>
<td>8</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>SPI</td>
<td>-</td>
</tr>
<tr>
<td>DP80390XP</td>
<td>on-chip RAM 64k/64k</td>
<td>on-chip ROM 256</td>
<td>8</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>SPI</td>
<td>-</td>
</tr>
</tbody>
</table>
PERFORMANCE

The following table gives a survey about the Core area and performance in Programmable Logic Devices after Place & Route (CPU features and peripherals included):

<table>
<thead>
<tr>
<th>Device</th>
<th>Speed grade</th>
<th>Area</th>
<th>F(_{\text{max}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLEX10KE</td>
<td>-1</td>
<td>2250 LC</td>
<td>57 MHz</td>
</tr>
<tr>
<td>ACEX1K</td>
<td>-1</td>
<td>2250 LC</td>
<td>57 MHz</td>
</tr>
<tr>
<td>APEX20KE</td>
<td>-1</td>
<td>2250 LC</td>
<td>50 MHz</td>
</tr>
<tr>
<td>APEX20KEC</td>
<td>-7</td>
<td>2250 LC</td>
<td>66 MHz</td>
</tr>
<tr>
<td>APEX-III</td>
<td>-7</td>
<td>2250 LC</td>
<td>78 MHz</td>
</tr>
<tr>
<td>CYCLONE</td>
<td>-6</td>
<td>2250 LC</td>
<td>76 MHz</td>
</tr>
<tr>
<td>CYCLONE-II</td>
<td>-6</td>
<td>2250 LC</td>
<td>91 MHz</td>
</tr>
<tr>
<td>CYCLONE-III</td>
<td>-6</td>
<td>2400 LC</td>
<td>111 MHz</td>
</tr>
<tr>
<td>Arria GX</td>
<td>-6</td>
<td>1595 ALUT</td>
<td>112 MHz</td>
</tr>
<tr>
<td>STRATIX</td>
<td>-5</td>
<td>2250 LC</td>
<td>90 MHz</td>
</tr>
<tr>
<td>STRATIX-II</td>
<td>-3</td>
<td>1580 ALUT</td>
<td>160 MHz</td>
</tr>
<tr>
<td>STRATIX-III</td>
<td>-2</td>
<td>1580 ALUT</td>
<td>191 MHz</td>
</tr>
<tr>
<td>STRATIX-IV</td>
<td>-2</td>
<td>1580 ALUT</td>
<td>196 MHz</td>
</tr>
</tbody>
</table>

Core performance in INTEL FPGA® devices – results given for working system with connected IDATA, CODE and XDATA memories

For the user, the most important aspect is an application speed improvement. The most commonly used arithmetic functions and their improvement are shown in the table below. The improvement was computed as \(\frac{\text{80C51 clock periods}}{\text{DP8051 clock periods}}\) required to execute an identical function. More details are available in the core documentation.

<table>
<thead>
<tr>
<th>Function</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit addition (immediate data)</td>
<td>9.00</td>
</tr>
<tr>
<td>8-bit addition (direct addressing)</td>
<td>9.00</td>
</tr>
<tr>
<td>8-bit addition (indirect addressing)</td>
<td>9.00</td>
</tr>
<tr>
<td>8-bit subtraction (register addressing)</td>
<td>12.00</td>
</tr>
<tr>
<td>8-bit subtraction (immediate data)</td>
<td>9.00</td>
</tr>
<tr>
<td>8-bit subtraction (direct addressing)</td>
<td>9.00</td>
</tr>
<tr>
<td>8-bit subtraction (indirect addressing)</td>
<td>9.00</td>
</tr>
<tr>
<td>8-bit subtraction (register addressing)</td>
<td>12.00</td>
</tr>
<tr>
<td>8-bit multiplication</td>
<td>16.00</td>
</tr>
<tr>
<td>8-bit division</td>
<td>9.60</td>
</tr>
<tr>
<td>16-bit addition</td>
<td>12.00</td>
</tr>
<tr>
<td>16-bit subtraction</td>
<td>12.00</td>
</tr>
<tr>
<td>16-bit multiplication</td>
<td>13.60</td>
</tr>
<tr>
<td>32-bit addition</td>
<td>12.00</td>
</tr>
<tr>
<td>32-bit subtraction</td>
<td>12.00</td>
</tr>
<tr>
<td>32-bit multiplication</td>
<td>12.60</td>
</tr>
<tr>
<td>Average speed improvement</td>
<td>11.12</td>
</tr>
</tbody>
</table>

Dhrystone Benchmark Version 2.1 was used to measure the core performance. The following table shows the DP8051 performance in terms of VAX MIPS per 1 MHz rating.

<table>
<thead>
<tr>
<th>Device</th>
<th>DMIPS/MHz</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>80C51</td>
<td>0.00941</td>
<td>1.00</td>
</tr>
<tr>
<td>DP8051</td>
<td>0.10787</td>
<td>11.46</td>
</tr>
<tr>
<td>DP8051+DPTRs</td>
<td>0.13722</td>
<td>14.58</td>
</tr>
<tr>
<td>DP8051+DPTRs+SXDM</td>
<td>0.14457</td>
<td>15.36</td>
</tr>
<tr>
<td>DP8051+DPTRs+SXDM+MDU32</td>
<td>0.14632</td>
<td>15.55</td>
</tr>
</tbody>
</table>

Core performance in terms of DMIPS per MHz