

### General Description

The Digital Blocks DB8051C-SP (Standard Peripherals) Microcontroller Verilog IP Core is compliant with the MCS@51 Instruction Set and contains standard 8051 MCU peripherals, including an interrupt controller, UART, two 16-bit timers, and four 8-bit I/O ports. The DB8051C-SP builds on the highly successful DB8051C CPU Core, utilizing Digital Blocks Micro-Architecture, which results in a small VLSI footprint and low Clocks Per Instruction performance.

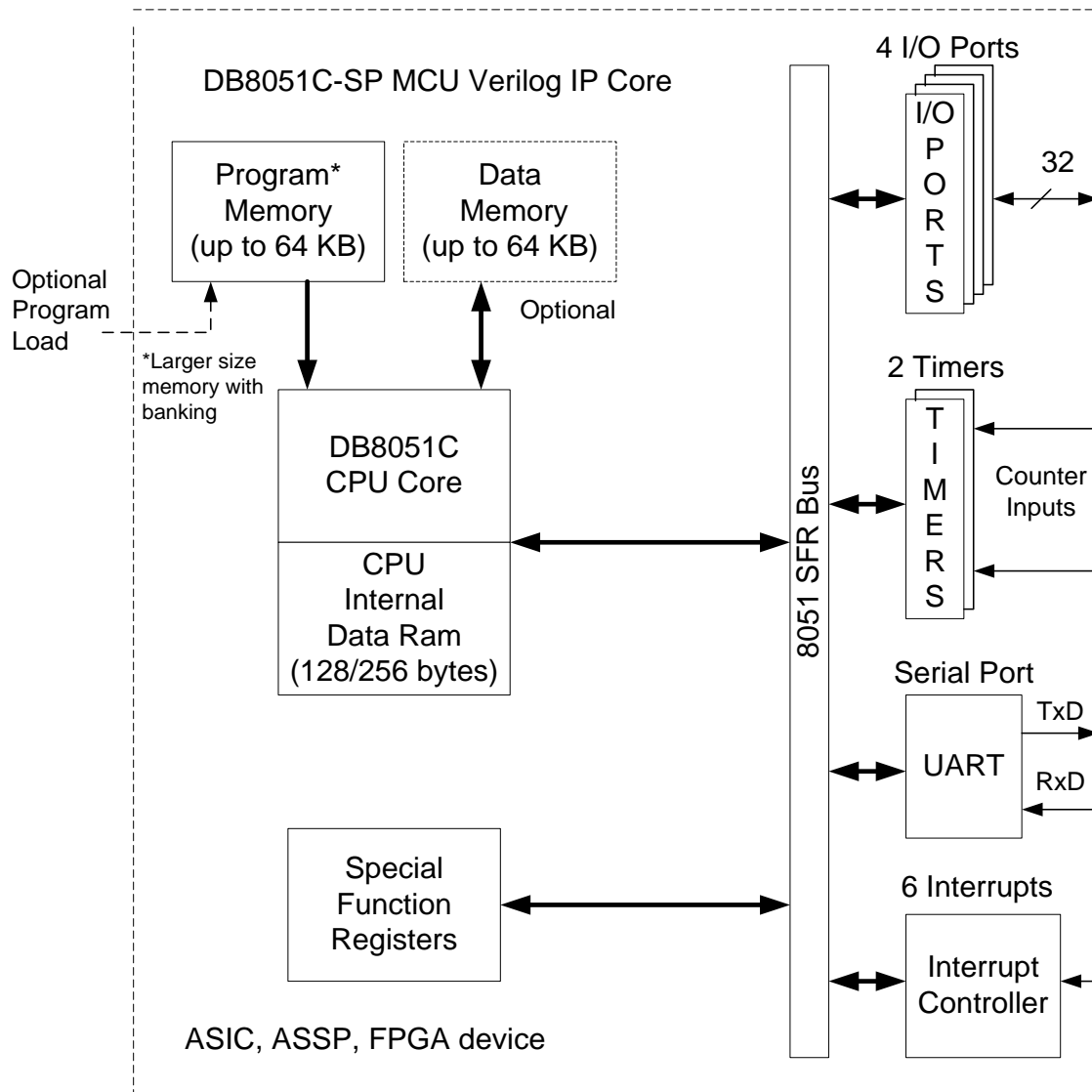


Figure 1: DB8051C-SP MCU Verilog IP Core Diagram

The DB8051C-SP CPU Core executes instructions out of Program Memory, sized by the user up to 64 KB. Higher Program Memory size is available through banking. Program Memory can be either non-volatile memory or SRAM loaded from a non-volatile memory source. Integrated within the DB8051C-SP CPU Core is 128/256 byte SRAM, for fast program data storage & manipulation. Optionally, the user can add up to 64 KB of Data Memory for larger data processing requirements.

The 8051 Microcontroller Verilog IP Core is an 8-bit architecture with 255 instructions, which include Arithmetic, Logical, Boolean Bit Manipulation, Data Transfer, & Program Branching. The Boolean Processor is for single-bit data manipulation, and the Arithmetic Unit's multiply/divide instructions consume a very low VLSI footprint. Combined with the 8051 Special Function Register interface (which optionally can be replaced with a AMBA APB interface), users can customize the peripherals beyond the standard peripherals, according to their unique requirements.

Digital Blocks Micro-Architecture achieves a minimum 3 Clocks Per Instruction (CPI) execution cycle. This achieves a higher throughput without resorting to the complex, higher resource requirements of pipelining, yielding a 4x speed-up CPI over the original industry standard MCS®51 microcontrollers, with an ultra-low VLSI utilization.

The 8051 Architecture with Register, Direct Address, Index Address, & Immediate Data addressing leads to smaller program space requirements. Thus, the DB8051C-SP 8-bit Microcontroller Verilog IP Core is an excellent choice for Embedded Control & Communications applications.

## Features

- 8-bit Microcontroller Binary Compatible with MCS®51 Instruction Set
- Standard 8051 Architecture
  - Arithmetic / Logical Unit
  - Hardware Multiply / Divide with low VLSI footprint
  - Boolean Processor for Bit Manipulation
  - Branching Instructions for Program Control
  - Data Transfer Instructions
  - 5 Addressing Modes: Register, Direct, Indirect, Immediate, Indexed. Requires less Program Code Memory
  - 111 instructions
  - Internal Data RAM, user defined as 128 or 256 bytes
- Enhanced 8051 Architecture
  - Minimum 3 Cycles Per Instruction Execution
- Streamlined ASIC & ASSP & FPGA Integration
  - Synchronous Memories
  - Hardware Micro Control Unit
  - Unidirectional Busses
  - Fully Static, Rising Edge Only, Synchronous Design
- Program Memory
  - User Defined, up to 64 KB. Larger size with Memory Banking
  - Non-Volatile Memory or Configurable SRAM
- Data Memory
  - User Defined, up to 64 KB
- Standard Peripherals:
  - Four 8-bit Ports for I/O
  - UART Controller
  - Two 16-bit Timer
  - Interrupt Controller
- Optional Peripherals (Please see Digital Blocks DB8051C-XP data sheet):
  - More I/O Ports, Serial Port Controllers, Timers (including a Watchdog Timer), and more interrupt sources
  - Serial Port Interface (SPI)
  - I2C Controller (I2C)
  - PWM unit with programmable frequency
  - Keyboard Hardware Controller
  - LCD Controller

**8051 Instruction Set Summary**

<b>Arithmetic Operation</b>			
<b>Instruction</b>		<b>Description</b>	<b>Bytes</b>
ADD	A, Rn	Add register to Accumulator	1
ADD	A, direct	Add direct byte to Accumulator	2
ADD	A, @Ri	Add indirect RAM to Accumulator	1
ADD	A,#data	Add immediate data to Accumulator	2
ADDC	A, Rn	Add register to Accumulator with Carry	1
ADDC	A, direct	Add direct byte to Accumulator with Carry	2
ADDC	A, @Ri	Add indirect RAM to Accumulator with Carry	1
ADDC	A,#data	Add immediate data to Accumulator with Carry	2
SUBB	A, Rn	Subtract Register from Accumulator with Borrow	1
SUBB	A, direct	Subtract direct byte from Accumulator with Carry	2
SUBB	A, @Ri	Subtract indirect RAM from Accumulator with Carry	1
SUBB	A,#data	Subtract immediate data from Accumulator with Carry	2
INC	A	Increment Accumulator	1
INC	Rn	Increment Register	1
INC	direct	Increment direct byte	2
INC	@ri	Increment direct RAM	1
DEC	A	Decrement Accumulator	1
DEC	Rn	Decrement Register	1
DEC	direct	Decrement direct byte	2
DEC	@ri	Decrement direct RAM	1
INC	DPTR	Increment Data Pointer	1
MUL	AB	Multiply A & B	1
DIV	AB	Divide A & B	1
DA	A	Decimal Adjust Accumulator	1

<b>Logical Operation</b>			
<b>Instruction</b>		<b>Description</b>	<b>Bytes</b>
ANL	A, Rn	AND Register to Accumulator	1
ANL	A, direct	AND direct byte to Accumulator	2
ANL	A, @Ri	AND indirect RAM to Accumulator	1
ANL	A,#data	AND immediate data to Accumulator	2
ANL	direct, A	AND Accumulator to direct byte	2
ANL	direct, #data	AND immediate data to direct byte	3
ORL	A, Rn	OR Register to Accumulator	1
ORL	A, direct	OR direct byte to Accumulator	2
ORL	A, @Ri	OR indirect RAM to Accumulator	1
ORL	A,#data	OR immediate data to Accumulator	2
ORL	direct, A	OR Accumulator to direct byte	2
ORL	direct, #data	OR immediate data to direct byte	3
XRL	A, Rn	Exclusive-OR Register to Accumulator	1
XRL	A, direct	Exclusive-OR direct byte to Accumulator	2
XRL	A, @Ri	Exclusive-OR indirect RAM to Accumulator	1
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2
XRL	direct, A	Exclusive-OR Accumulator to direct byte	2
XRL	direct, #data	Exclusive-OR immediate data to direct byte	3
CLR	A	Clear Accumulator	1
CPL	A	Complement Accumulator	1
RL	A	Rotate Accumulator Left	1
RLC	A	Rotate Accumulator Left through the Carry	1
RR	A	Rotate Accumulator Right	1
RRC	A	Rotate Accumulator Right through the Carry	1
SWAP	A	Swap Nibbles within the Accumulator	1

<b>Data Transfer Operation</b>			
<b>Instruction</b>		<b>Description</b>	<b>Bytes</b>
MOV	A, Rn	Move Register to Accumulator	1
MOV	A, direct	Move direct byte to Accumulator	2
MOV	A, @Ri	Move indirect RAM to Accumulator	1
MOV	A, #data	Move immediate data to Accumulator	2
MOV	Rn, A	Move Accumulator to Register	1
MOV	Rn, direct	Move direct byte to Register	2
MOV	Rn, #data	Move immediate data to Register	2
MOV	direct, A	Move Accumulator to direct byte	2
MOV	direct, Rn	Move Register to direct byte	2
MOV	direct, direct	Move direct byte to direct byte	3
MOV	direct, @Ri	Move indirect RAM to direct byte	2
MOV	direct, #data	Move immediate data to direct byte	3
MOV	@Ri, A	Move Accumulator to indirect RAM	1
MOV	@Ri, direct	Move direct byte to indirect RAM	2
MOV	@Ri, #data	Move immediate data to indirect RAM	2
MOV	DPTR, #data16	Load Data Pointer with a 16-bit constant	3
MOVC	A, @A+DPTR	Move Code byte relative to DPTR to Accumulator	1
MOVC	A, @A+PC	Move Code byte relative to PC to Accumulator	1
MOVX	A, @Ri	Move External Data RAM (8-bit Address) to Accumulator	1
MOVX	A, @DPTR	Move External Data RAM (16-bit Address) to Accumulator	1
MOVX	@Ri, A	Move Accumulator to External Data RAM (8-bit Address)	1
MOVX	@DPTR, A	Move Accumulator to External Data RAM (16-bit Address)	1
PUSH	direct	Push direct byte onto stack	2
POP	Direct	Pop direct byte from stack	2
XCH	A, Rn	Exchange Register with Accumulator	1
XCH	A, direct	Exchange direct byte with Accumulator	2
XCHD	A, @Ri	Exchange low-order Digit indirect RAM with Accumulator	1

<b>Boolean Variable Manipulation</b>			
<b>Instruction</b>		<b>Description</b>	<b>Bytes</b>
CLR	C	Clear Carry	1
CLR	bit	Clear direct bit	2
SETB	C	Set Carry	1
SETB	bit	Set direct bit	2
CPL	C	Complement Carry	1
CPL	bit	Complement direct bit	2
ANL	C, bit	AND direct bit to Carry	2
ANL	C, /bit	AND complement of direct bit to Carry	2
ORL	C, bit	OR direct bit to Carry	2
ORL	C, /bit	OR complement of direct bit to Carry	2
MOV	C, bit	Move direct bit to Carry	2
MOV	bit, C	Move carry to direct bit	2
JC	rel	Jump if Carry is set	2
JNC	rel	Jump if Carry not set	2
JB	bit, rel	Jump if direct bit is set	3
JNB	bit, rel	Jump if direct bit is Not set	3
JBC	Bit, rel	Jump if direct bit is set & clear bit	3

<b>Program Branching</b>			
<b>Instruction</b>		<b>Description</b>	<b>Bytes</b>
ACALL	addr11	Absolute Subroutine Call	2
LCALL	addr16	Long Subroutine Call	3
RET		Return from Subroutine	1
RETI		Return from interrupt	1
AJMP	addr11	Absolute Jump	2
LJMP	addr16	Long Jump	3
SJMP	rel	Short Jump (relative addr)	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1
JZ	rel	Jump if Accumulator is zero	2
JNZ	rel	Jump if Accumulator is Not Zero	2
CJNE	A, direct, rel	Compare direct byte to Accumulator and Jump if Not Equal	3
CJNE	A, #data, rel	Compare immediate to Accumulator and Jump if Not Equal	3
CJNE	Rn, #data, rel	Compare immediate to register and Jump if Not Equal	3
CJNE	@Ri, #data, rel	Compare immediate to indirect and Jump if Not Equal	3
DJNZ	Rn, rel	Decrement Register and Jump if Not Zero	2
DJNZ	direct, rel	Decrement direct byte and Jump if Not Zero	3
NOP		No Operation	1



## Verification Method

Digital Blocks provides Simulation & FPGA verification platforms. The DB8051C-SP simulation test suite loads Program Memory, exercises all instructions, and checks expected results. Building out from the directed tests are 8051 programs loaded and run on the DB8051C-SP with expected results checked.

The DB8051C-SP FPGA verification platform exercises the DB8051C-SP in Xilinx & Altera FPGAs. Please contact Digital Blocks for additional information.

## Customer Evaluation

Digital Blocks offers a variety of methods for prospective customers to evaluate the DB8051C-SP. Please contact Digital Blocks for additional information.

## Deliverables

The DB8051C-SP is available in synthesizable RTL Verilog, along with the simulation test bench with expected results, datasheet, and user manual.

## Ordering Information

Please contact Digital Blocks for additional technical, pricing, evaluation, and support information.

Digital Blocks, Inc.  
PO Box 192  
587 Rock Rd  
Glen Rock, NJ 07452 USA  
Phone: +1-201-251-1281  
Fax: +1-702-552-1905  
[info@digitalblocks.com](mailto:info@digitalblocks.com)

Copyright © Digital Blocks, Inc. 2009-2017, ALL RIGHTS RESERVED

###

Digital Blocks is a registered trademark of Digital Blocks, Inc.  
All other trademarks are the property of their respective owners