



Fraunhofer Institut
Integrierte Schaltungen

***CorePool* FHG8051**

Databook

- subject to change without notice -

Version: v1.5

Date: 02.08.02

Document History

Table 1: Document History

Version	Date	Responsible	Description
v1.0	12.01.97	Shn	Generic datasheet
v1.1	17.02.97	Shn	Add timing diagrams
v1.2	29.02.97	Shn	Detail register description
v1.3	24.04.97	Shn	Enhance interrupt unit
v1.4	14.11.97	Shn	Detail timing diagrams
v1.5	20.01.98	Shn	Detail interrupt behaviour, update document structure

Contact

CorePool
Fraunhofer Institute Integrated Circuits

Am Wolfsmantel 33
91058 Erlangen
Germany

Phone: +49 (0) 9131 776 777
Fax: +49 (0) 9131 776 499
Email: info@corepool.com
Internet: <http://www.corepool.com>

Table of Contents

Document History	2
Contact	2
Purpose	7
Features	7
Design Kit	7
Requirements	7
References	8
Block Diagram	8
Signal Description	9
General Information	9
Parameter Description	10
VHDL Usage trough Component Instantiation	13
Verilog Usage trough Component Instantiation	16
Architecture	18
Functional Description	20
Control Unit	27
Interfaces	28
Waveforms and timing tables	29
Instruction Set	37
Opcode list	39
Application notes	48

List of Tables and Figures

Document History	2
Block Diagram	8
Signal Description	9
Parameter Description	10
Architecture	18
Internal Memory Map	20
Program Status Word	21
SFR Memory Map	21
Configuration Register	22
Register Bank Selector Bits	22
Power Control Register	23
Interrupt Priority Register	24
Interrupt Enable Register	25
Interrupt Request Sequence	25
Reset Sequence	27
Waveform of program memory read access	29
Timing table of program memory read access	29
Waveform of program memory write access	30
Timing table of program memory write access	30
Waveform of external data memory read access	31
Timing table of external data memory read access	31
Waveform of external data memory write access	32
Timing table of an external data memory write access	32
Waveform of internal data memory read and write accesses	33
Timing table of internal data memory read and write accesses	33
Waveform of SFR read and write accesses	34
Timing table of SFR read and write accesses	34
Waveform of idle mode	35
Timing table of idle mode	35

List of Tables and Figures (cont)

Arithmetic Instructions 37

Logical Instructions 37

Data Transfer Instructions 37

Bit Manipulation Instructions 38

Program Control Instructions 38

Opcode List 40

Application Example 48

Purpose

The FHG8051 is a high performance, opcode compatible core version of the industry standard 8051 micro controller for ASIC and FPGA implementations. Its parameter set enables the designer to enhance the cpu performance and to tailor the architecture to the applications needs.

Features

- 8-Bit Synthesizeable Microcontroller Core
- Opcode and Cycle Equivalent to Industry Standard 8051
- Fully Static, Microcode Free Design
- Up to 64K Bytes Program Memory Address Space
- Up to 64K Bytes Data Memory Address Space
- Up to 256 Bytes Internal Data Memory
- Up to 128 Special Function Registers (SFR)
- Idle, Power Down Mode
- Programmable Clock and Wait State Generator
- Optional up to 7 Interrupt Sources in 2 Priority Levels or No Interrupt Unit
- Optional True 8051 Cycle Operation or Reduced Cycle Mode for Enhanced Performance
- Optional PCON, P1, P3 or Second DPTR Register
- Optional Area Reducing NOP Behaviour for MUL, DIV, DA Opcodes
- Optional Program Memory Write Mode

Design Kit

- Technology Independent Implementation as Synopsys Design Ware Components
- VHDL/Verilog Simulation Models
- VHDL/Verilog 8051 Compliance Test Suite
- Auxiliary Simulation Models for User Testbenches
- Synthesis and Testsynthesis Scripts
- Example Design and Testchip available
- Design Support, Netlist Synthesis Service and Consulting available

Requirements

Simulation

- VHDL IEEE 1076 Simulator (Synopsys VSS, Modeltech VSIM, Vantage, others)
- Verilog IEEE Simulator (Cadence VerilogXL, Modeltech VSIM, others)

Synthesis

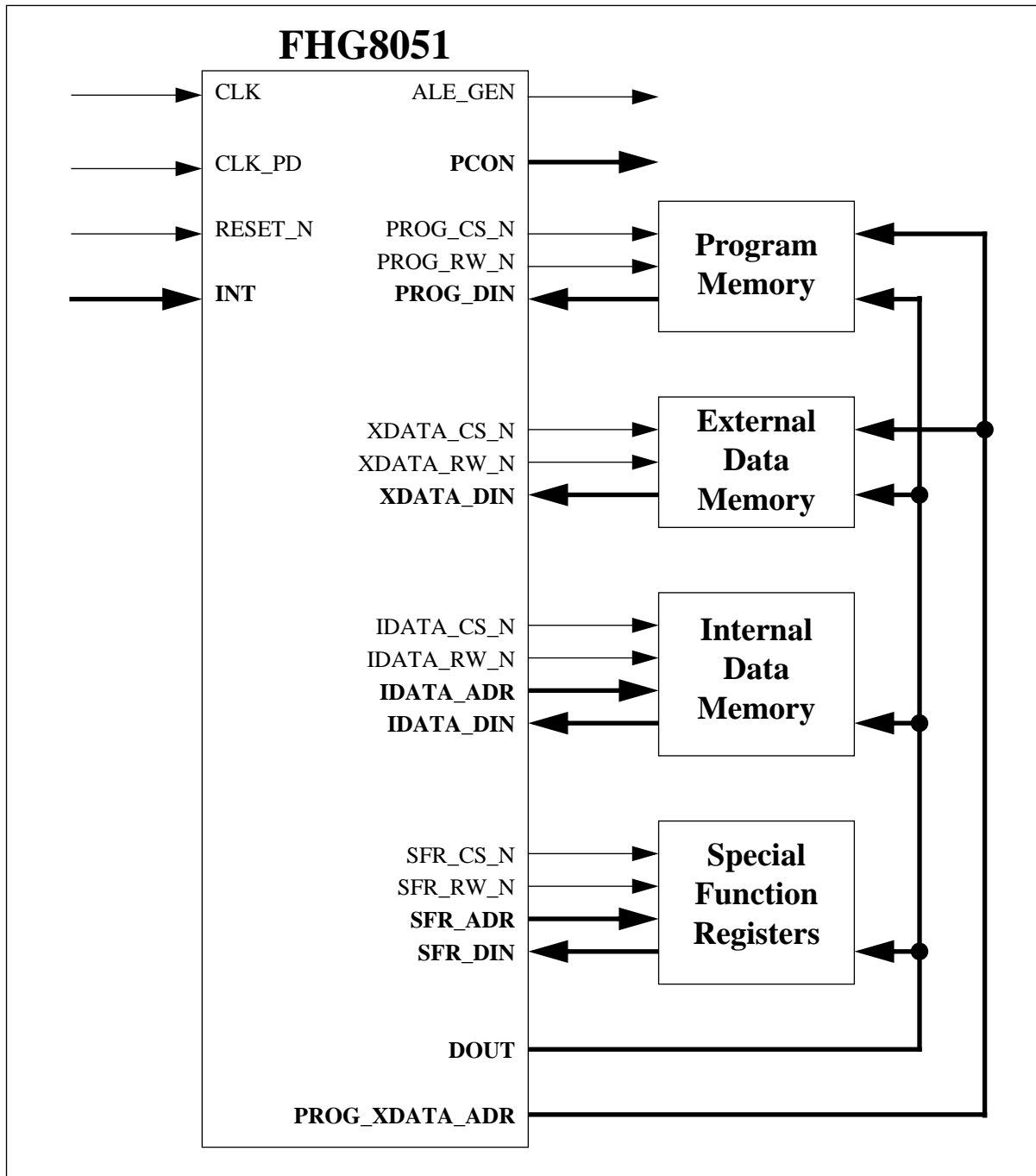
- Synopsys Design Compiler

References

For more details about the 8051 cpu refer to an industry standard databook about this micro controller architecture.

Block Diagram

Figure 1: Block Diagram



Signal Description

Table 2: Signal Description

Pin	Direction	Bitwidth	Description
CLK	IN	1	Systemclock.
CLK_PD	IN	1	Power down systemclock.
RESET_N	IN	1	Low activ asynchronous reset.
INT	IN	7	High active Interrupt bus for 7 independent interrupt sources.
ALE_GEN	OUT	1	High active address latch enable.
PCON	OUT	8	High active signal for power down control.
PROG_CS_N	OUT	1	Low active chip select for program memory.
PROG_RW_N	OUT	1	Low active read/write for program memory. High level indicates a read, low level a write operation.
PROG_DIN	IN	8	Input databus bus from program memory.
PROG_XDATA_ADR	OUT	16	Addressbus for external data memory and program memory.
XDATA_CS_N	OUT	1	Low active chip select for external memory.
XDATA_RW_N	OUT	1	Low active read/write for the external memory. High level indicates a read, low level a write operation.
XDATA_DIN	IN	8	Input databus from external memory.
IDATA_CS_N	OUT	1	Low active chip select for internal memory.
IDATA_RW_N	OUT	1	Low active read/write for internal memory. High level indicates a read, low level a write operation.
IDATA_ADR	OUT	8	Address bus for the internal memory.
IDATA_DIN	IN	8	Input databus from internal memory.
SFR_CS_N	OUT	1	Low active chip select for SFR memory.
SFR_RW_N	OUT	1	Low active read/write for SFR memory. High level indicates a read, low level a write operation.
SFR_ADR	OUT	7	Addressbus for SFR memory.
SFR_DIN	IN	8	Input databus from SFR memory.
DOUT	OUT	8	Common output databus to program memory (optional), external memory, internal memory and SFR memory.

General Information

External data memory and internal data memory means only the memory areas accessed by the 8051 instructions. It does not mean any onchip or offchip implementation of the memories. This is free to the designers needs.

Parameter Description

Tabelle 3: Parameter Description

Name	Type	Default Value	Value Range	Description
CYCLE_REDUCTION	Integer	0	{0,1}	Speed up instruction execution
IMPLEMENT_INT	Integer	5	{0...7}	Implement interrupt logic
IMPLEMENT_MUL	Integer	1	{0,1}	Implement logic for multiplication opcode
IMPLEMENT_DIV	Integer	1	{0,1}	Implement logic for division opcode
IMPLEMENT_DA	Integer	1	{0,1}	Implement logic for Decimal adjust opcode
IMPLEMENT_DPTR2	Integer	0	{0,1}	Implement second DPTR register
IMPLEMENT_PCON	Integer	1	{0...8}	Implement (larger) PCON register
IMPLEMENT_P1	Integer	0	{0,1}	Implement additional P1 register
IMPLEMENT_P3	Integer	0	{0,1}	Implement additional P3 register
IMPLEMENT_PWR	Integer	0	{0,1}	Implement program memory write function for MOVX opcode

The high speed architecture of the FHG8051 executes every instruction cycle in 6 clock cycles instead of 12 in a standard 8051. The parameter `CYCLE_REDUCTION` allows to speed up a set of accelerateable instructions from 2 instruction cycles to 1 additionally. Please refer to the opcode list and its comments to identify these instructions. By default `CYCLE_REDUCTION` is set to “0“ and the classic 2 instruction cycle execution will be built. When `CYCLE_REDUCTION` is set to “1“ the set of accelerateable instructions will execute in 1 instruction cycle and increase performance. Use this parameter to speed up execution if you don’t need a cycle identical behaviour of your existing assembler code.

The FHG8051 has an interrupt control unit which allows up to 7 different interrupt sources. By default parameter `IMPLEMENT_INT` ist set to “5“ like in the classic 8051 implementation, and all necessary logic for the interrupt unit will be built. When `IMPLEMENT_INT` ist set to “0“ no interrupt logic is built. Use this parameter if you want to reduce chip size and don’t need any interrupts.

The parameter `IMPLEMENT_MUL` enables the designer to reduce chip area by not implementing the necessary logic for the multiplication instruction `MUL`. By default `IMPLEMENT_MUL` is set to “1“ and the multiplication logic will be built. When `IMPLEMENT_MUL` is set to “0“ no logic will be built and the behaviour of `MUL` is like a `NOP` instruction. Use this parameter if you don’t need the multiplication instruction.

The parameter `IMPLEMENT_DIV` enables the designer to reduce chip area by not implementing the necessary logic for the division instruction `DIV`. By default `IMPLEMENT_DIV` is set to “1” and the division logic will be built. When `IMPLEMENT_DIV` is set to “0” no logic will be built and the behaviour of `DIV` is like a `NOP` instruction. Use this parameter if you don’t need the division instruction.

The parameter `IMPLEMENT_DA` enables the designer to reduce chip area by not implementing the necessary logic for the decimal adjust instruction `DA`. By default `IMPLEMENT_DA` is set to “1” and the decimal adjust instruction logic will be built. When `IMPLEMENT_DA` is set to “0” no logic will be built and the behaviour of `DA` is like a `NOP` instruction. Use this parameter if you don’t need the decimal adjust instruction.

With parameter `IMPLEMENT_DPTR2` the FHG8051 may contain a second `DPTR` register. This is useful e.g. for block transfers using `DPTR` and `DPTR2` as source and destination address registers. The standard `DPTR` registers are located at 82 h (`DPL`) and 83 h (`DPH`). The `DPTR2` registers are located at 84 h (`DPL2`) and 85 h (`DPH2`). To select `DPTR` or `DPTR2` as address source for operations the bit `DPSEL` will be used. It is located at the bit 0 of the `CFG` register at 86 h. If `DPSEL` is set to “0” `DPTR` is used, else `DPTR2`. The instructions affected by this feature are:

```
JMP @A+DPTR
INC DPTR
MOV DPTR, #16-Bit immediate
MOVX A, @DPTR
MOVX @DPTR, A
MOVC A, @A+DPTR
```

By default parameter `IMPLEMENT_DPTR2` is set to “0” and only the standard `DPTR` registers will be built. When `IMPLEMENT_DPTR2` is set to “1” the `DPTR2` registers and the `DPSEL` bit are built. Use this parameter if you want to enhance your address registers.

The parameter `IMPLEMENT_PCON` enables the designer to select the bitwidth of register `PCON` for application specific needs, e.g. power down bits for peripherals or custom units. By default `IMPLEMENT_PCON` is set to “1” and only bit 0 will be built. When `IMPLEMENT_PCON` is set to “0” no register will be built and all bits of output port `PCON` are set to “0”. If `IMPLEMENT_PCON` is set to n ($1 \leq n \leq 8$), n bits of the register will be built starting at bit 0. All not implemented bits of register `PCON` are set to “0” for output port `PCON`. Use this parameter if you need additional control bits for your application.

The classic 8051 has the two address registers `P0` and `P2`. With the parameters `IMPLEMENT_P1` and `IMPLEMENT_P3` additional registers `P1` and `P3` can be built. By default `IMPLEMENT_P1` and `IMPLEMENT_P3` are set to “0” and no register will be built. By setting each parameter to “1” the respective register will be built.

The parameter IMPLEMENT_PWR enables the program memory write function for MOVX instructions. By default IMPLEMENT_PWR is set to “0” and no program memory write function will be implemented. When set to “1” an additional control bit PWR in register CFG at bit 1 is built, controlling the program memory write function. By setting PWR to “0” the regular MOVX operation proceeds. Setting PWR to “1” enables the program memory write function. Any read operation with MOVX will read from the external data memory, but the write operation with MOVX will now write to the program memory. This is very useful for downloading some program code from the external data memory, if a part of the program memory is implemented as a RAM.