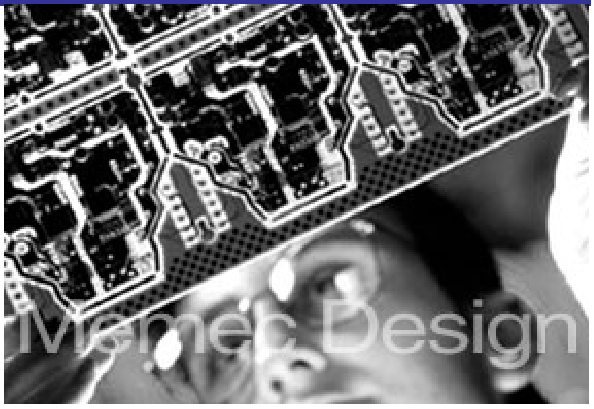
MC-ACT-6809

February 25, 2003

Software Compatible 6809 CPU

Datasheet v1.2



Actel CompanionCore

MemecCore™ Product Line
3721 Valley Centre Drive
San Diego, CA 92130 USA
Americas: +1 800-752-3040
Europe: +41 (0) 32 37432 00
Asia: +(852) 2410 2720
E-mail: actel.info@memecdesign.com
URL: www.memecdesign.com/actel

Product Summary

Intended Use

- Replacing existing 6809 designs
- Systemintegration of 6809 designs
- Designswith complex 8bit processors

Key Features

- 6809Software Compatible
- IllegalOpcode Recognition
- On-chipBus Concept
- FullySynchronous

Targeted Devices

- AcceleratorFamily
- ProASIC^{PLUS} Family
- SX-A/SX,RTSX Family

General Description

The MC-ACT-6809 is a softwarecompatible 6809 microprocessor implemented in VHDL using a structured and synchronous designmethodology. This flexible 8-bit microprocessor can be used for low to mid range applications and is easily integratedinto recent Actel FPGA technologies. The 6809 is designed that application specific performance enhancements(e.g., high-speed multiplier) can be integrated, too.

Memec Design built a simulation and ahardware verification environment. The simulation environment is VHDL based andsimulates the synthesizable VHDL code of the 6809.This functional VHDL testbench reading instruction code from a file can beused for performance check, software verification orto get usedto the functionality.

A demonstration board is available which shows the functionality of the 6809 in a hardware testbench.

Core Deliverables

- NetlistVersion
  - Netlistcompatible with the Actel Designer place and routetool
- RTLVersion
  - VHDLSource Code
  - TestBench
- All
  - UserGuide

Synthesis and Simulation Support

- Synthesis:Synplicity
- Simulation:ModelSim
- Othertools supported upon request

Verification

- TestBench

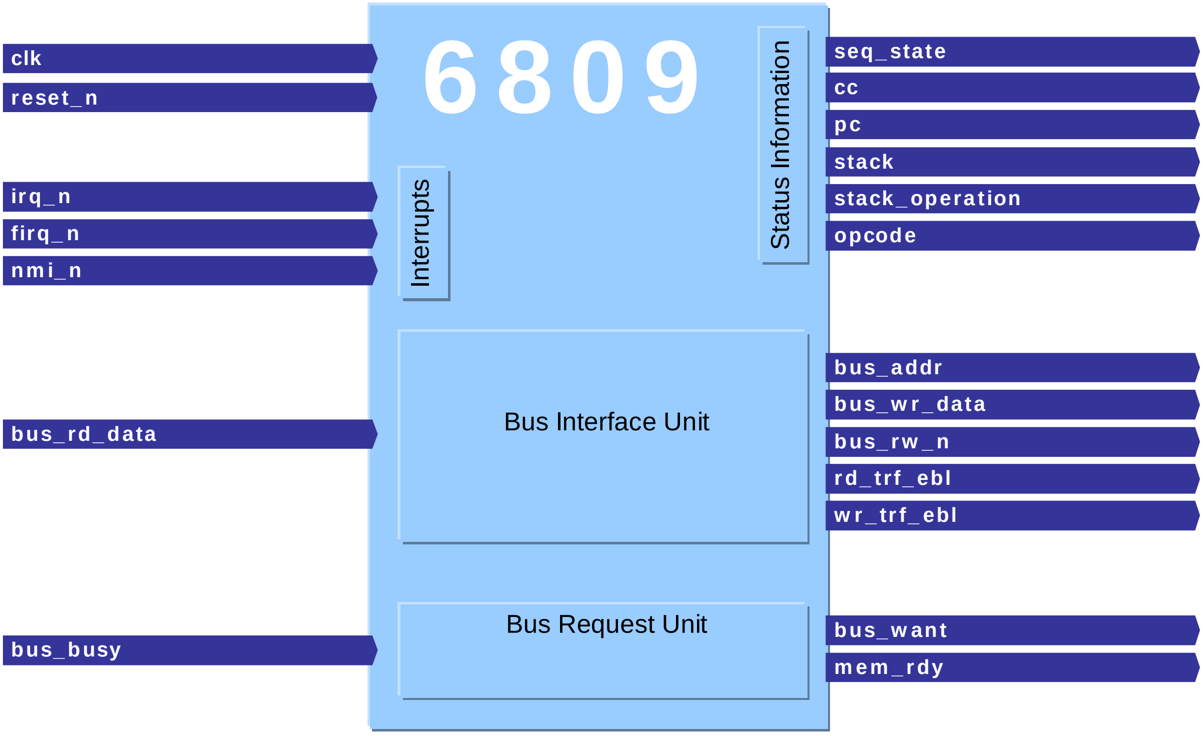


Figure 1: Logic Symbol

Functional Description

Structure of 6809

The block diagram below shows the index arithmetic unit, the data ALU, command sequencer, memory fetch unit and program counter.

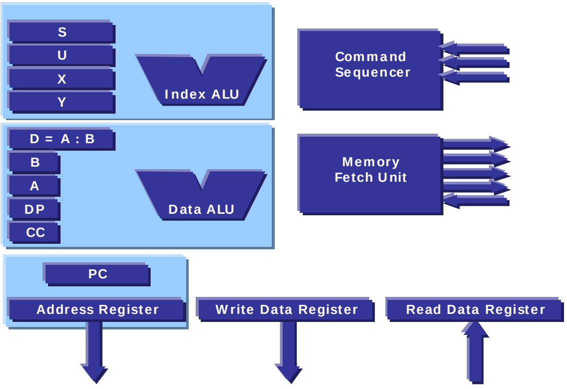


Figure 2: Block Diagram

The memory interface contains registered read and write data and address registers, which makes the design of the interface easier for timing verification etc.

Internal Registers

The following part describes the existing registers in the MC-ACT-6809:

Index Registers (X,Y)

The index registers are used in indexed mode of addressing and are used for calculating the effective addresses. During some indexed modes, the contents of the index register are incremented or decremented to point to the next item of tabular type data. All 4 pointer register (U,S, X, Y) can be used as index register.

Stack Pointer Registers (U,S)

The system stack pointer S is automatically used by the processor during subroutine calls and interrupts! The user stack pointer is controlled by the programmer. Both stack pointers point to the top of the stack.

Program Counter Register

The program counter (PC) points to the address of the next instruction to be executed.

Accumulators (A,B,D)

The A and B accumulators are general purpose registers. Some instructions concatenate the registers A and B to achieve 16 bit values. This register is then referred as the D register with the A register as most significant byte.

Condition Code (CC) Register

Beginning with bit(0) the condition code register contains following flags:

C:	Carry from ALU
V:	Overflow (2 complement) from ALU
Z:	Zero from ALU
N:	Negative (2 complement) from ALU
I:	Interrupt mask register for irq_n
H:	Half Carry (low nibble) from ALU
F:	Interrupt mask for firq_n
E:	Entire Flag for irq_n/firq_n stack operations

Direct Page (DP) Register

These registers enhance the direct addressing mode. During direct addressing instruction execution, the contents of DP appears at the higher address outputs (A15- A8).

Instructions

The MC-ACT-6809 supports all 6809 instructions like load, store, and, or, add, sub, transfer/exchange instructions for all registers, 8 by 8 unsigned multiply, decimal adjust and signed/unsigned conditional branches. A powerful stack command is also available. Software interrupts are supported.

Addressing Modes

Nearly all instructions can be combined with the following addressing modes:

Inherent: Register source and destination, no post bytes, e.g. INCA

Immediate: Postbyte is parameter, e.g. LDA #100

Extended: Postbytes are address, e.g. LDA 1000

Direct: Postbyte is low address, DP is high address, e.g. LDA <50

Indexed: Postbyte(s) define indexed mode:

- (Index registers are X,Y,U,S)
- Post incr. by 1 and 2, e.g. LDA ,S++
- Pre decr. by 1 and 2, e.g. LDA ,-X
- Zero offset, e.g. LDA 0,X
- signed offset, e.g. LDA -7,Y
- A/B/D signed offset, e.g. LDB A,U
- 8/16bit signed offset PC relative, e.g. LDA 665,PCR

Indexed indirect: The same types as indexed may be used. The indirect mode uses the argument of the index mode as address for an indirect access.

Illegal Instructions

To insure save operation, illegal opcodes are recognized and will lead to an exception, where a vector is fetched to enter the exception handler routine. For this purpose, the reserved vector of the 6809 is used. The address of the illegal instruction is pushed on the S stack.

Interrupts

For interrupting theMC-ACT-6809, there are3 level-sensitive interrupts:

irq_n: standardinterrupt or interrupt request, low_active, pushes all registers on stack:
PC(Low), PC(High), U(Low), U(High), Y(Low), Y(High), X(Low),X(High), DP, B, A, CC

firq_n: fastinterrupt, pushes onlyPC and CC on stack: PC(Low), PC(High), CC

nmi_n: nonmaskable interrupt, pushes all registers on stack:
PC(Low),PC(High), U(Low), U(High), Y(Low), Y(High), X(Low),X(High), DP, B, A, CC
The nmi_n is masked after reset until the system stackpointer is initialized.

The different interrupts will fetch the correspondingvector for the interruptin the memory:

0xfff0	IllegalInstruction
0xfff2	SWI3
0xfff4	SWI2
0xfff6	FIRQ
0xfff8	IRQ
0xfffa	SWI
0xfffc	NMI
0xfffe	RESET

The interrupt behavior ofthe MC-ACT-6809 matches exactly the one of the original 6809 in terms of results in registers andmemory. An exception is the case where an illegal instruction is decoded by the 6809: in this case, the reserved interrupt isused.

Device Requirements

Family	Device	Utilization			Performance
		COMB	SEQ	Total	
SX-A	SX32A-3	1625 (91%)	344 (12%)	1969(69%)	28MHz
ProASIC^{PLUS}	APA150-STD	n/a	n/a	2795 (56%)	11MHz
Axcelerator	AX500-3	1577 (30%)	339 (13%)	1916(24%)	31MHz

Table 1: Device Utilization and Performance

Verification and Compliance

Functional and timing simulation has been performed on the 6809 using VHDL and Verilog Test Benches. Simulation vectors used for verification are provided with the core. This core has also been used successfully in customer designs.

Signal Descriptions

The following signal descriptions define the IO signals.

Signal	Direction	Description
clk	in	system clock, rising edge used only
reset_n	in	asynchronous system reset, active low, goes to all flip flops
nmi_n	in	Non maskable interrupt, active low
firq_n	in	Fast Interrupt request, active low
irq_n	in	Interrupt request, active low
bus_busy	in	Bus busy, If '1' the uP can not access the bus and waits until bus is released.
wr_trf_ebl	out	Write transfer enable
bus_want	out	Bus want. The uP requests the bus.
rd_trf_ebl	out	Read transfer enable
rw_n	out	Read / write-not. If '1' then read cycle
bus_addr[15:0]	out	Bus address
bus_rd_data[7:0]	in	Read data
bus_wr_data[7:0]	out	Write data
mem_rdy	out	Memory ready, see 'Internal Bus Interface Timing' for timing diagram
seq_state[5:0]	out	Command sequencer state register
cc[7:0]	out	Condition code register
pc[15:0]	out	Program counter register
stack[15:0]	out	Stack pointer
stack_operation	out	Stack operation, if '1' a stack operation cycle is running
opcode[5:0]	out	Operation code

Table 2: Core I/O Signals

Recommended DesignExperience

For the source version, users shouldbe familiar with HDL entry and Actel design flows. Users should be familiar with Actel Libero v2.2 Integrated Design Environment (IDE) and preferably with Synplify and ModelSim.

Ordering Information

Part Number	Description
MC-ACT-6809-NET	CoreNetlist
MC-ACT-6809-VHD	CoreVHDL

Table 3: Core Part Numbers

The CORE is provided underlicense from Memec Design for use in Actel programmable logic devices. Please contact Memec Design for pricing andmore information.

Information furnished byMemec Design is believed to be accurate and reliable.Memec Design reserves the right to change specificationsdetailed in this data sheet at any time without notice, in order to improve reliability, function or design, andassumes no responsibility for any errors within this document. Memec Design does notmake any commitment to update this information.

Memec Design assumes no obligation to correct any errors contained herein orto advise any user ofthis textof any correction, if such be made, nor does the Company assume responsibility for thefunctioning of undescribed features orparameters. Memec Design will not assume anyliability for theaccuracy or correctness of any support or assistance providedto a user.

Memec Design does not representhat products described herein are free frompatent infringement or from any other third-party right. No license is granted byimplication or otherwise under anypatent or patent rights ofMemec Design.

MemecCore products are not intendedfor use in life support appliances, devices, or systems. Use of aMemecCore product in such application withoutthe written consent of the appropriateMemec Design officer is prohibited.

All trademarks, registeredtrademarks, or service marks are property of theirrespective owners.

Datasheet Revision History

Version	Date	Description
Datasheet 1.0	December19, 2002	First Release
Datasheet 1.1	January23, 2003	Modification done in section core deliverables, Added logo to footer
Datasheet 1.2	February 25, 2003	Modification done in section device requirements, new URL and address inserted