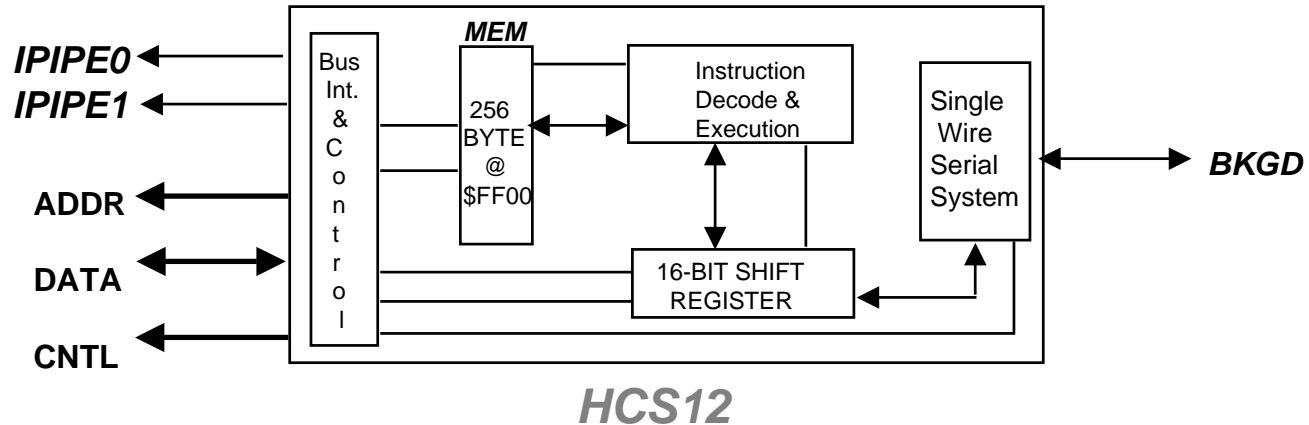


S12 Micro Controller

Debug Module & BDM Tools

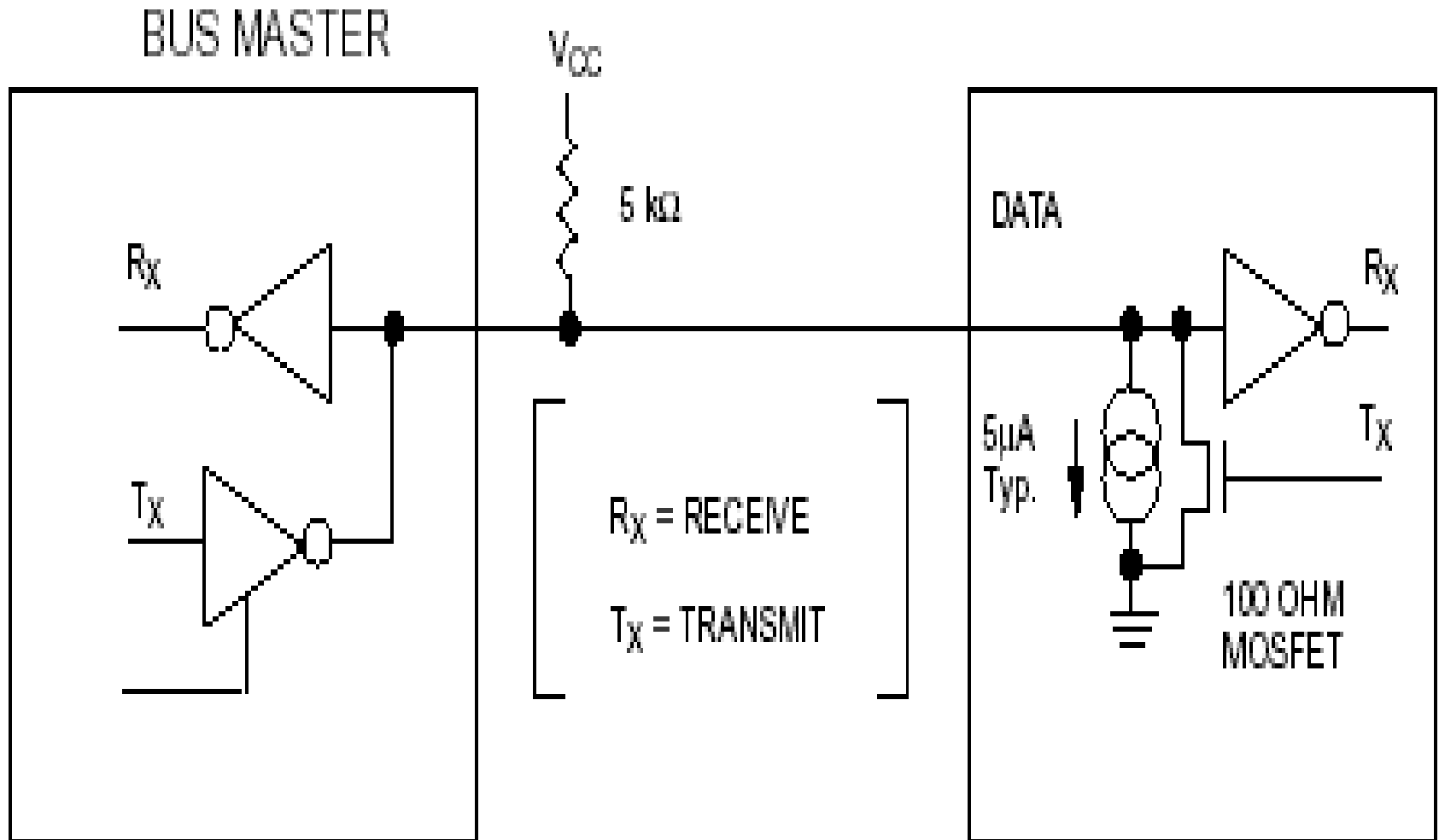
Debug Mode



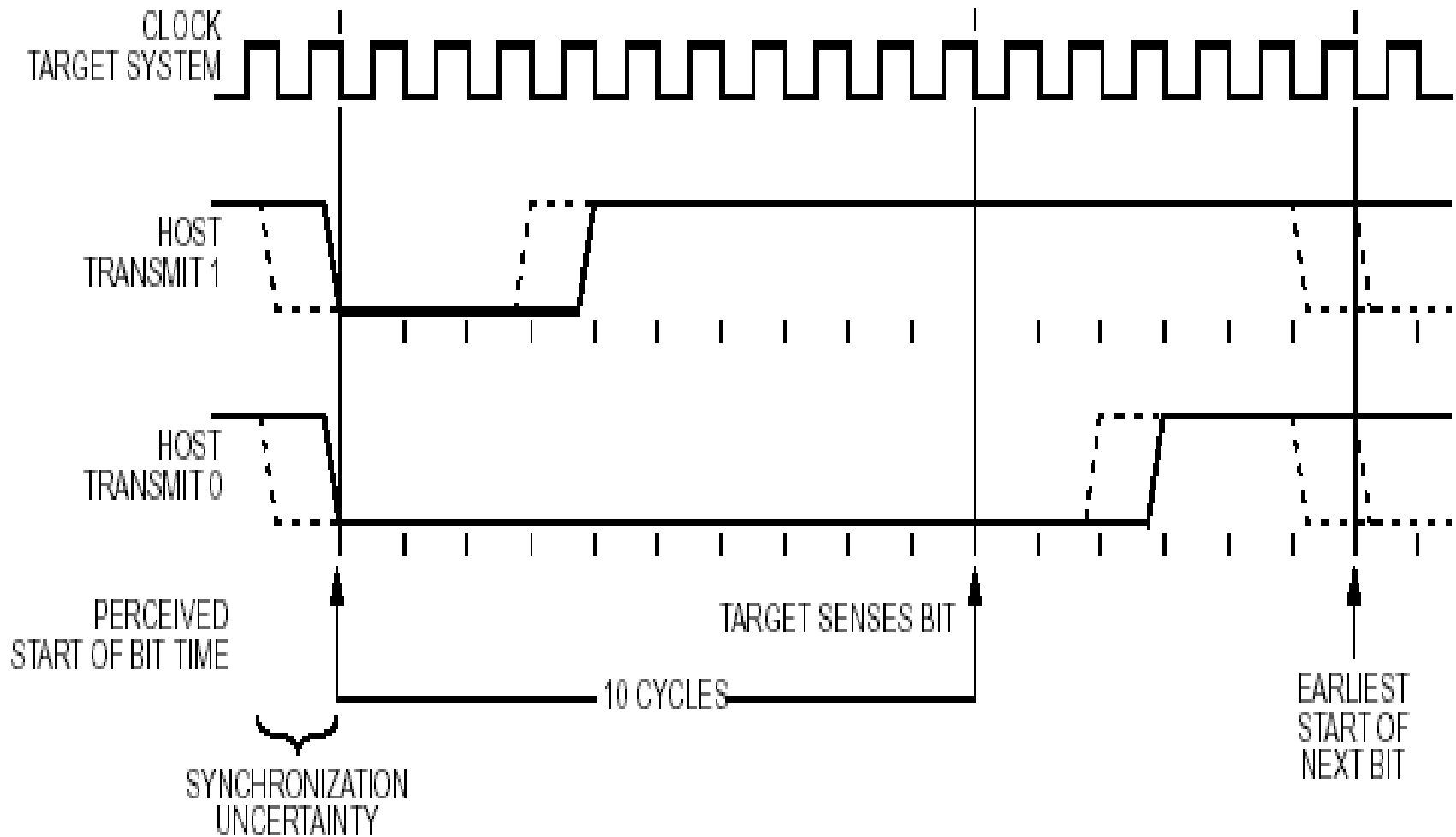
Features:

- Single-Wire Interface.
- Active out of Reset in **Special Single-Chip Mode**
- Instruction Tagging Capability
- Allows for system debugging **during program execution.**
- Can access **all CPU** registers and system **memory.**

1 wire communication

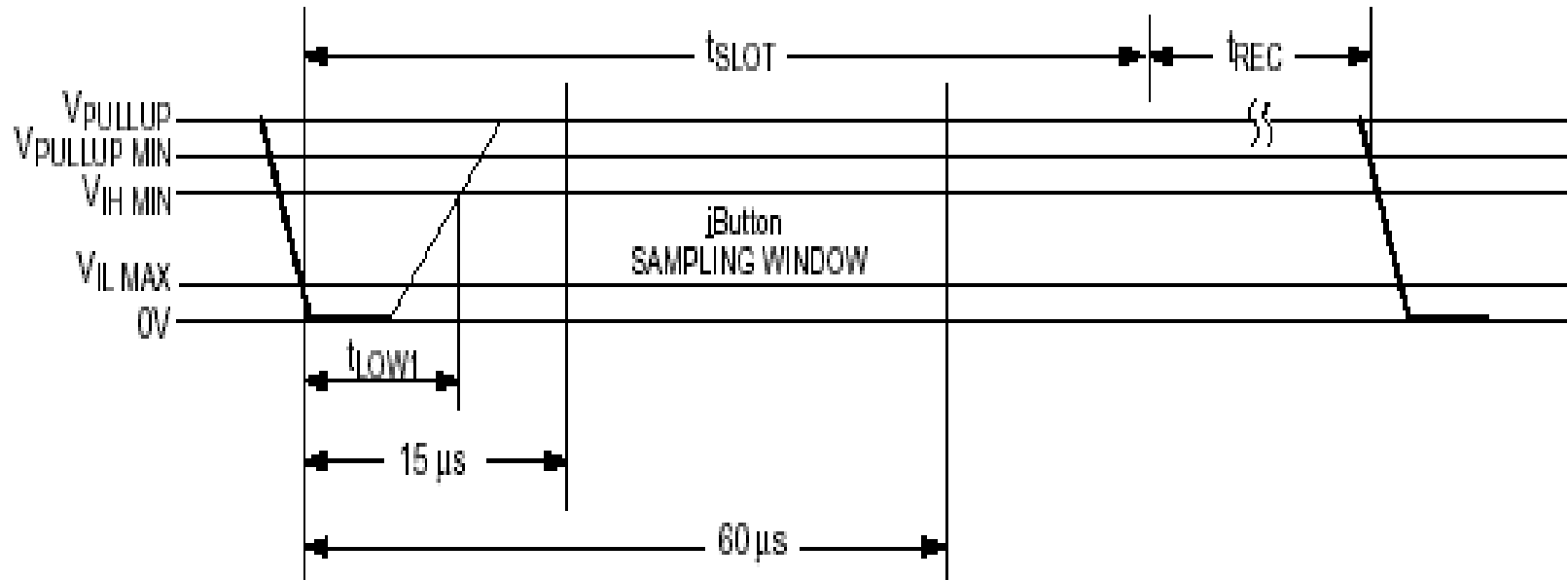


BDM Host-to-Target Serial Bit Timing



Write One Time slot

Write-One Time Slot



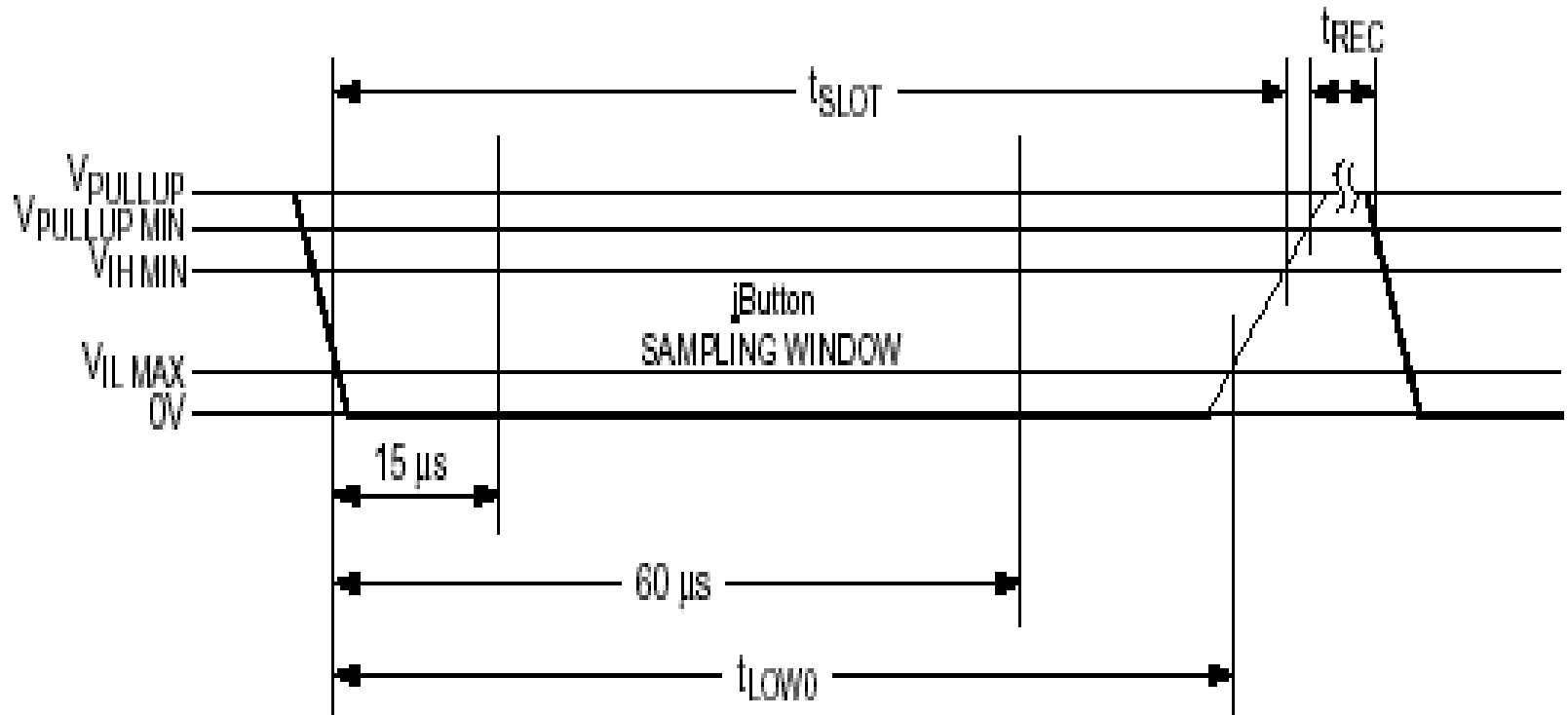
$$60\ \mu s \leq t_{SLOT} < 120\ \mu s$$

$$1\ \mu s \leq t_{LOW1} < 15\ \mu s$$

$$1\ \mu s \leq t_{REG} < \infty$$

Write Zero Time slot

Write-Zero Time Slot

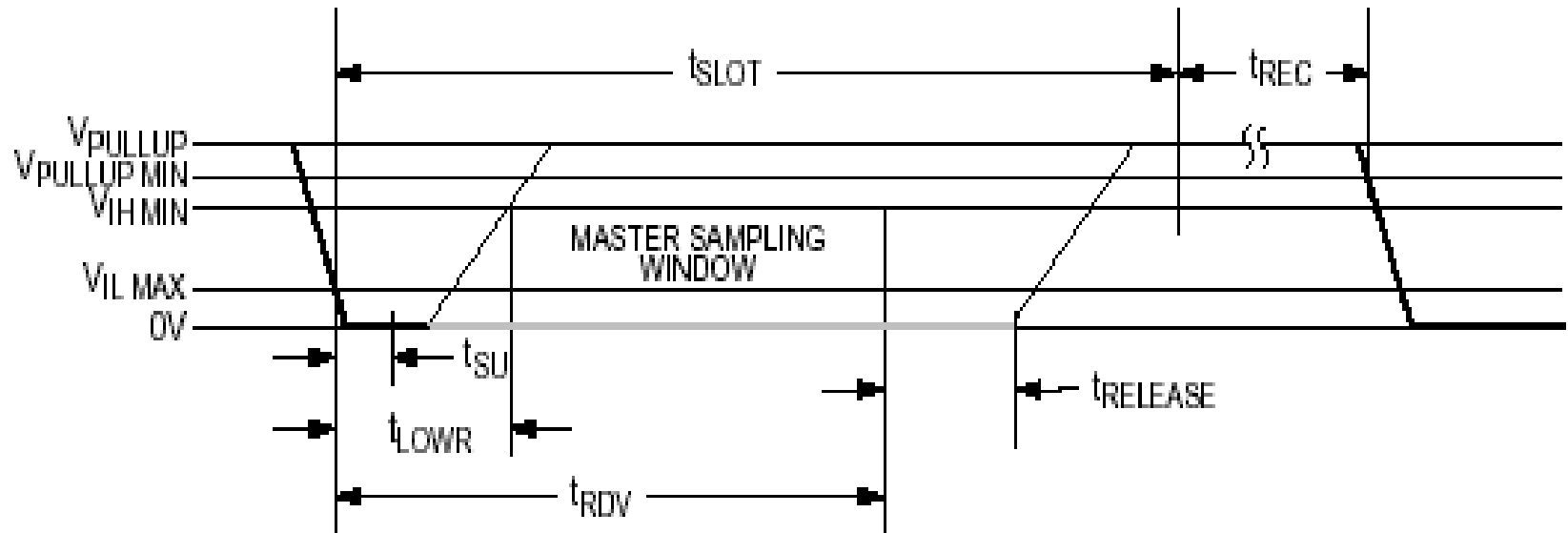


$$60 \mu s \leq t_{LOW0} \leq t_{SLOT} \leq 120 \mu s$$

$$1 \mu s \leq t_{REC} \leq \infty$$

Read Data

Read-Data Time Slot



— RESISTOR
— MASTER

$$60 \mu s \leq t_{SLOT} < 120 \mu s$$

$$1 \mu s \leq t_{LOWR} < 15 \mu s$$

$$0 \leq t_{RELEASE} < 45 \mu s$$

$$1 \mu s \leq t_{REC} < \infty$$

$$t_{RDV} = 15 \mu s$$

$$t_{SU} < 1 \mu s$$

Write a Byte in Register B to 1 word Bus

LSLB



BCC WRITE0

or

BCS WRITE1

Write a Byte in Register B to BKGD

| | | | |
|-----------|------|----------------|-----------------------|
| WRITEBYTE | LDAA | #\$8 | ; Bit number |
| NEXTBIT | LSLB | | ; High bit to C |
| | BCC | WRITE0 | |
| | BCLR | BDMPORT,BKGD | ; Write 1, DDR=OUTPT |
| | SEI | | ; Disable IRQ |
| | BSET | BDMDDR,BKGD | ; 0 |
| | NOP | | |
| | BSET | BDMPORT,BKGD | ; to 1 |
| NEXTBIT1 | BCLR | BDMDDR,BKGD | ; 3 cycles, DDR=Input |
| | CLI | | ; Enable IRQ |
| | DECA | | ; 1 bit sent |
| | BNE | NEXTBIT | |
| WAIT10 | RTS | | ; 5Cycles |
| WRITE0 | BSET | BDMDDR,BKGD | ; DDR = Output |
| | SEI | | |
| | BCLR | BDMPORT,BDMBIT | ; Output = 0 |
| | BSR | WAIT10 | ; 5 + 4 cycles |
| | NOP | | |
| | BRA | NEXBIT1 | ; 3 cycles |

Background Debug Mode

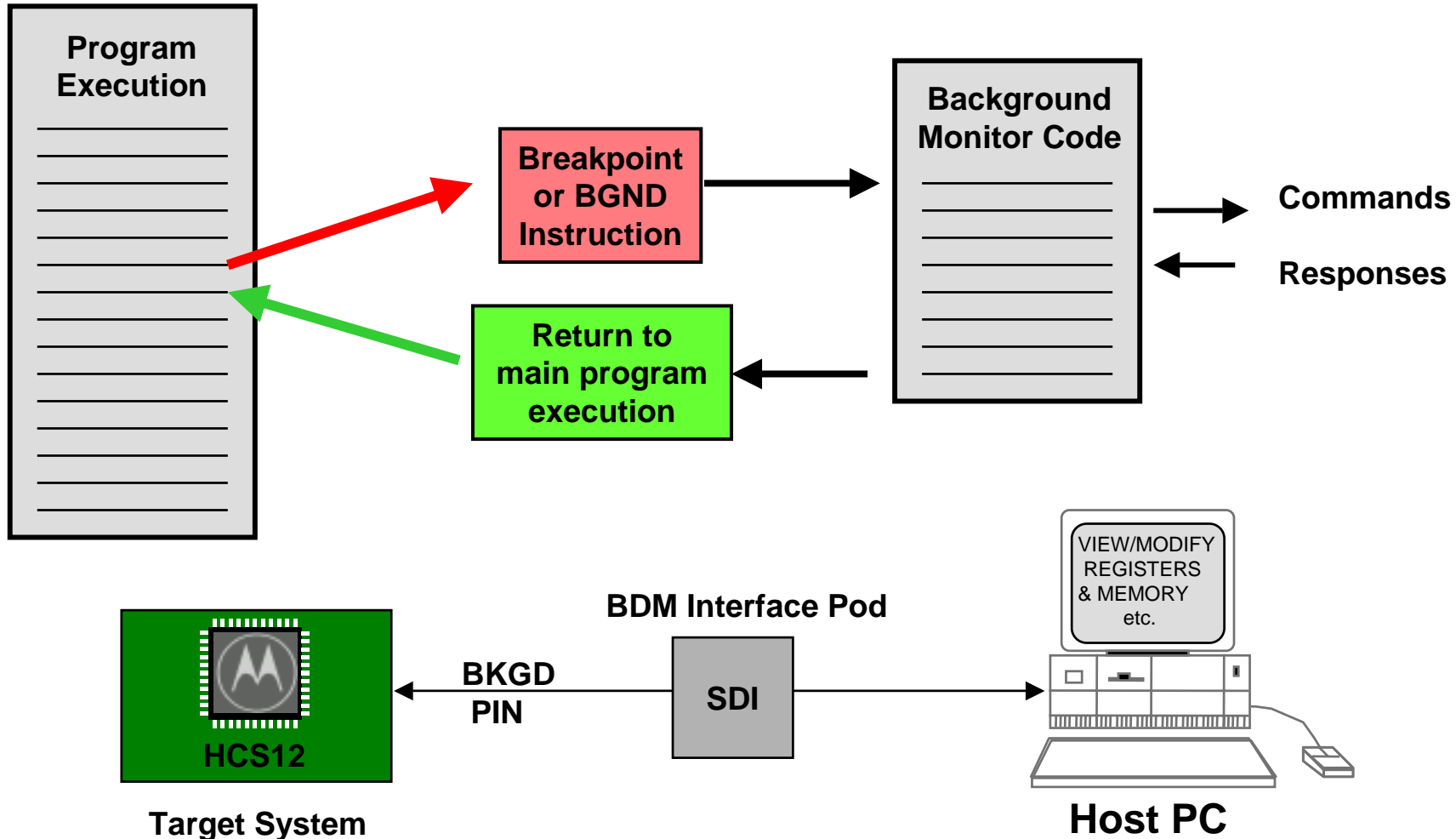
Once enabled, background mode can be made active by a serial command sent via the BKGD pin or execution of a CPU12 BGND instruction.

The CPU executes code located in a small on-chip ROM mapped to addresses \$FF20 to \$FFFF, replaces the regular system vectors .

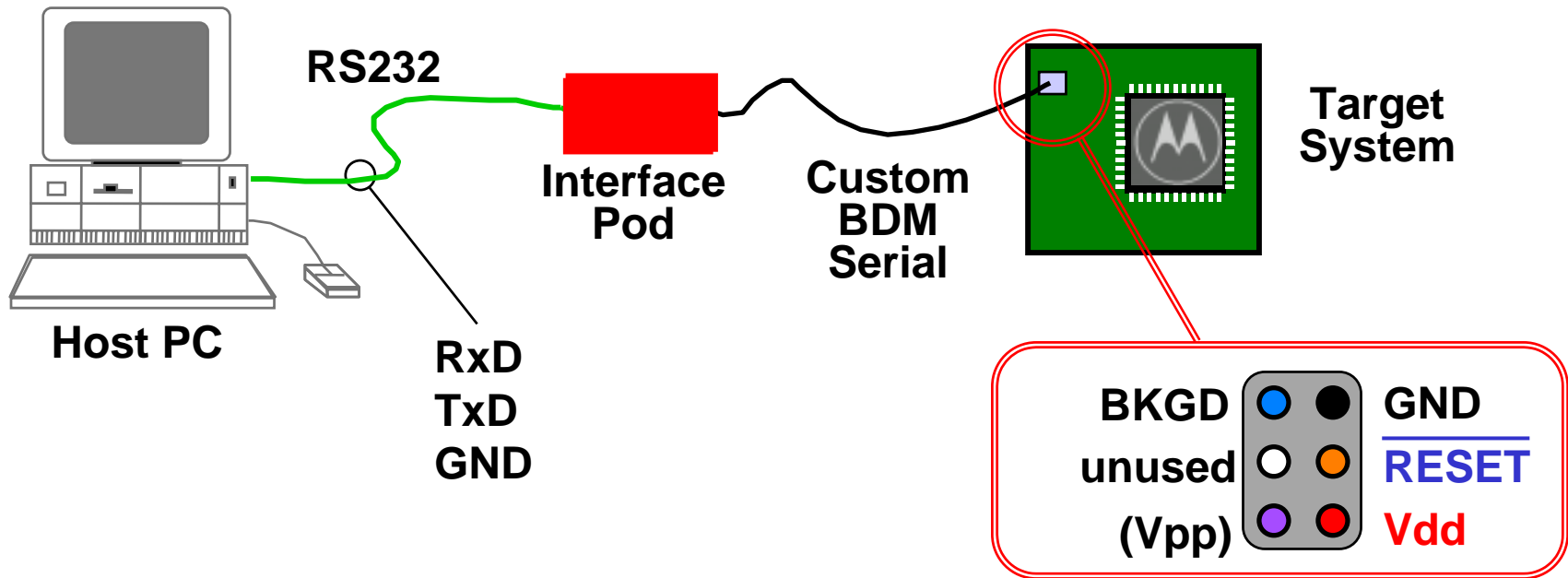
BDM control registers are at \$FF00 to \$FF06.
The user memory from \$FF00 to \$FFFF is not in the map except through serial BDM commands.

Background Debug Mode

Single Wire Development Interface



Typical BDM System

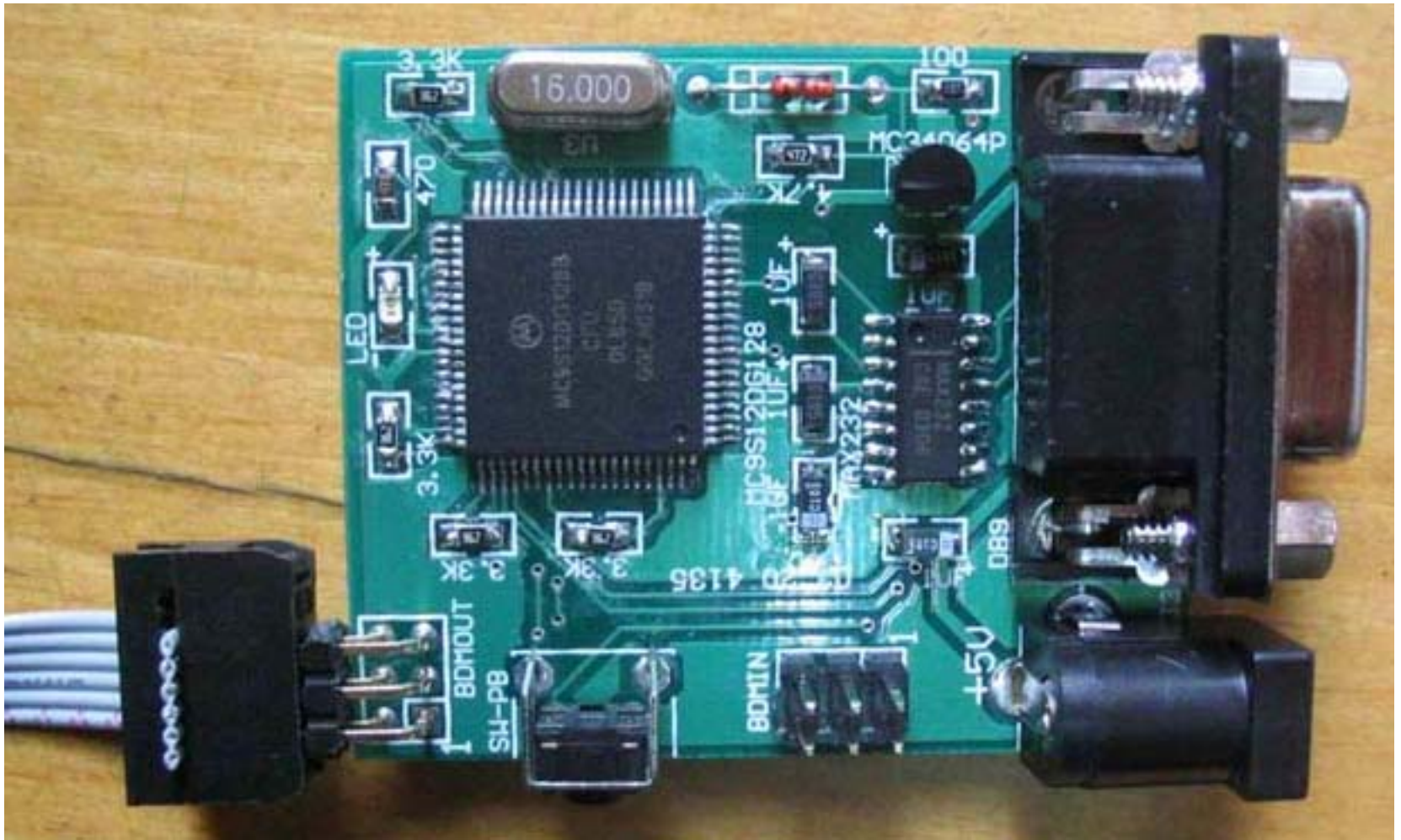


Only BKGD and GND are required, others are optional.

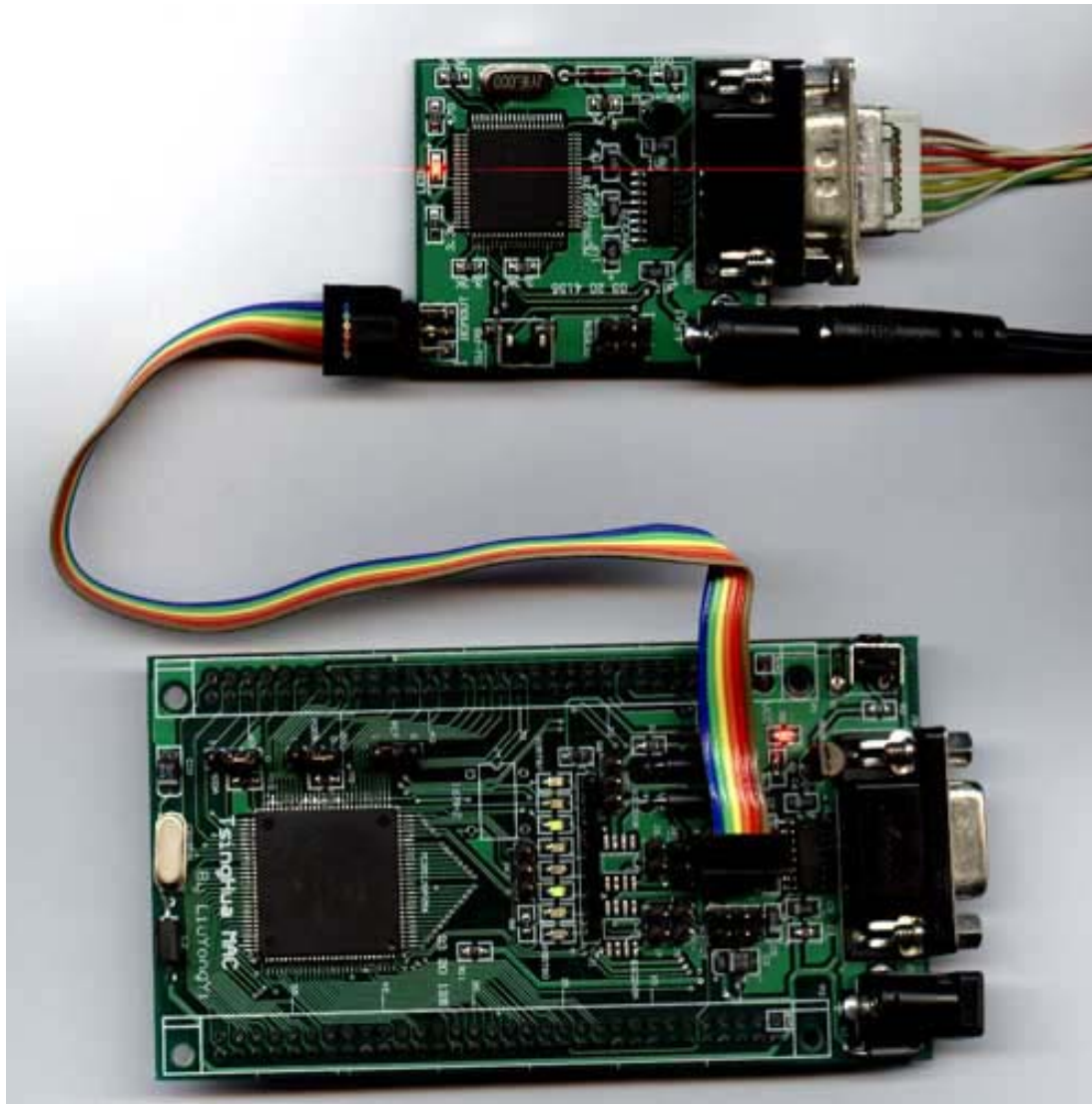
- Reset would allow host to reset the target system.
- Vdd would allow pod to “steal” power from the target system.
- Vpp is not needed in any current HCS12 systems.

注意：一定不能插反！！！！

S12 DG128 Based BDM Tool



Use BDM Tool Debug S12 Target System



BKGD Pin Shares 3 Functions

Mode Select during reset

Selects Normal (BKGD=1) or Special (BKGD=0) modes
During Reset

normal single chip (run mode) or
special single chip (development mode)

Serial communication with BDM

Pseudo Open-Drain, Bi-direction

Data transfer rates up to 300 kbits/second (including overhead)

Tagging (for tagging instructions as they are fetched into the pipe)

Tagging and serial BDM commands are mutually exclusive

BDM Multilink Pod

The new **BDM MULTILINK** enables you to make higher-speed downloads to your target. The cable derives its power from the target and can operate with targets from 5.5V down to 2 Volts. **BDM MULTILINK** also automatically calculates the speed and voltage of the target. It will work with the latest (and future!) Motorola CPU12 devices, up to 35MHz internal frequency.



BDM Hardware Commands

| COMMANDS | OPCODE | DISCRIPTION |
|-----------------|--------|------------------------------------|
| BACKGROUND | 90 | Enter Background Mode |
| READ_BD_BYTE | E4 | Read a Byte with BDM in Map |
| READ_BD_WORD | EC | Read a Word with BDM in Map |
| READ_BYTE | E0 | Read a Byte with BDM out of Map |
| READ_WORD | E8 | Read a Word with BDM out of Map |
| WRITE_BD_BYTE | C4 | Write a Byte with BDM in Map |
| ENABLE_FIRMWARE | C4 | Write Byte \$ff01 to Set the ENBDM |
| WRITE_BD_WORD | CC | Write a Word with BDM in Map |
| WRITE_BYTE | C0 | Write a Byte with BDM out of Map |
| WRITE_WORD | C8 | Write a Word with BDM out of Map |

BDM Firmware Commands

| COMMANDS | OPCODE | DISCRIPTION |
|------------|--------|--|
| READ_NEXT | 62 | Read next Word Pointed to by X |
| READ_PC | 63 | Read Program Counter |
| READ_D | 64 | Read D Accumulator |
| READ_X | 65 | Read X Index Register |
| READ_Y | 66 | Read Y Index Register |
| READ_SP | 67 | Read Stack Pointer |
| WRITE_NEXT | 42 | Write next Word Pointed to by X |
| WRITE_PC | 43 | Write Program Counter |
| WRITE_D | 44 | Write D Accumulator |
| WRITE_X | 45 | Write X Index Register |
| WRITE_Y | 46 | Write Y Index Register |
| WRITE_SP | 47 | Write Stack Pointer |
| GO | 08 | Go to User Program |
| TRACE1 | 10 | Trace One Instruction then return to BDM |
| TAGG0 | 18 | Enable Tagging and Go to User Program |

BDM Registers

| Address | Register |
|----------------------|--|
| \$FF00 | BDM instruction register |
| \$FF01 | BDM status register |
| \$FF02–\$FF03 | BDM shift register |
| \$FF04–\$FF05 | BDM address register |
| \$FF06 | BDM condition code register (CCR) |

\$FF01 BDM status register

| | | | | | | | |
|-------|--------|-------|-----|-------|-------|-------|---|
| ENBDM | BDMACT | ENTAG | SDV | TRACE | CLKSW | UNSEC | 0 |
|-------|--------|-------|-----|-------|-------|-------|---|

| | |
|--------|---|
| ENBDM | Enable BDM |
| BDMACT | BDM active status |
| ENTAG | Tagging enable |
| SDV | Shift data valid |
| TRACE | TRACE1 BDM firmware command is being executed |
| CLKSW | Clock switch |
| UNSEC | Unsecure |

\$FF00 BDM Instruction Register

| | | | | | | | |
|------------|-------------|------------|--------------|------------|-------------|----------|----------|
| H/F | DATA | R/W | BKGND | W/B | BD/U | 0 | 0 |
|------------|-------------|------------|--------------|------------|-------------|----------|----------|

| | |
|--------------|-------------------------------------|
| H/F | Hardware/firmware flag |
| DATA | Data follow flag |
| R/W | Read/write flag |
| BKGND | Enter active background mode |
| W/B | Word/byte transfer flag |
| BD/U | BDM map/user map flag |

Standard BDM Firmware Listing(1)

*** EQUATES**

fff6 BDMVEC equ\$fff6 ;First BDM ROM vector.

ff00 org\$ff00 ;Start of BDM map (registers)

ff00 INSTR rmb1 ;Instruction (command) register

*** s/w ! H/S ! DATA ! R/W ! TTAG : GO ! R2 ! R1 ! R0 !**

*** hdw ! H/S ! DATA ! R/W !BKGND : W/B !BD/USR! NEXT ! - !**

*** Reg codes: R2:R1:R0**

*** 0:0:0 - Illegal, command \$00 is null command**

*** 0:0:1 - not used**

*** 0:1:0 - Next Word 2,+X pre inc X by 2 and r/w next word (,X)**

*** later r/w next will work from ADDRESS reg value not X**

*** 0:1:1 - PC**

*** 1:0:0 - D**

*** 1:0:1 - X**

*** 1:1:0 - Y**

*** 1:1:1 - SP**

*** TTAG:GO coding:**

*** 0:0 - No execution command**

*** 0:1 - Go to user program**

*** 1:0 - Trace one user instruction and return to BDM**

Standard BDM Firmware Listing(2)

orgff20 ;BDM ROM start

AFTER_RST ;***CAUTION 7. *****CAUTION 8.**

| | | | |
|--------------|-----------------------|-----------------------------------|--|
| ff20 | 1c ff 01 80 | bset STATUS \$80 | ;Set the ENBDM bit to pass the brset |
| START | | | |
| ff24 | b7 b4 | exgt3 d | ;Save D without affecting CCR. |
| ff26 | b7 20 | tfrccr a | |
| ff28 | 7a ff 06 | staa CCRSAVE | ;Save user CCR value |
| ff2b | b7 d3 | exgx t2 | ;pc into x. *****CAUTION 4. |
| ff2d | 8e ff 00 | cpx #\$FF00 | ;Check to see if user PC overlaps BDM |
| | | | ;ROM. |
| ff30 | 24 04 | bhsROM_INC | ;If so, increment regardless. |
| ff32 | e7 00 | tst0,x | ;Test next opcode. This instruction |
| ff34 | 26 01 | bneRES_X_T2 | ;if not \$00, restore |
| ff36 | 08 ROM_INC | inx | ;else inc, then restore. This |
| ff37 | b7 d3 | exgx t2 | ;restore pc to temp 2 |
| ff39 | 1e ff 01 80 06 | brsetSTATUS \$80 INST_LOOP | ;Check if BDM allowed |
| ff3e | 87 | clra | ;Exit if BDM not allowed |
| ff3f | 20 1c | braEXIT_SEQ | |

Read: HCS12 V1.5 Core User Guide Version 1.2

BDM Conclusions

Much simpler connections than Logic Analyzers or Emulators

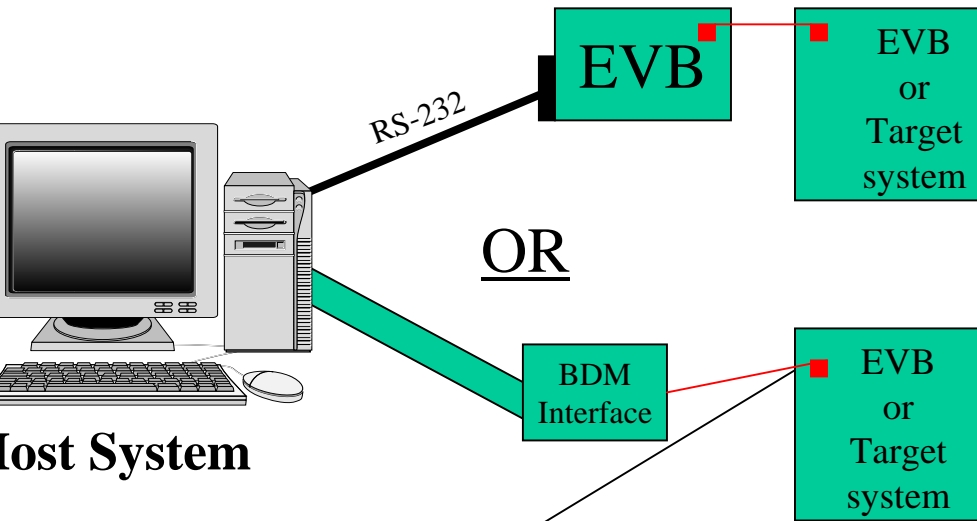
- **Less Cost**
- **Allows Debug Under REAL Operating Conditions**
- **No Need to Turn Off System to Connect to BDM**

Read/Write Access Without Disturbing the Running Programs

BDM Used for More Than Just Debug

- **Code download (EEPROM, Flash, RAM, external memory devices)**
- **Product maintenance without disassembly**
- **Calibration and monitoring system operation (non-intrusively)**

HC12 BDM Set-up Diagram

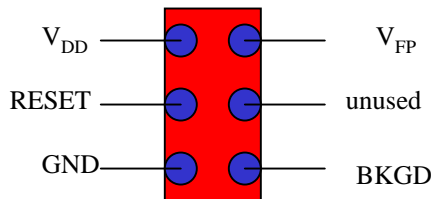


EVB in “pod mode”:

- Lowest cost BDM solution
 - \$99 EVB: M68EVB912B32
- Uses monitor program on MCU
- Does not allow for C-source level debugging

BDM Interface:

- Motorola and 3rd party hardware support
 - Motorola part number: M68SDIL12
- \$600 - \$2000 solution depending on vendor and feature set
- Most are compliant with all debuggers including GUI
- Allows for C-source level debugging



Motorola BDM connection: 6-pin header

- Easily added to target or production system
- Allows real-time debugging of the actual MCU used in the product without sacrificing MCU functions
- Eliminates the need for expensive and cumbersome emulators

HCS12 Commercial Development Tools

HCS12 Development Tools

Evaluation Board



HCS12 EVB NOW

www.metrowerks.com

Compiler/Debugger



cx6812 / ZAP 6812
NOW

www.cosmic-software.com

BDM Debugger



JProbeHC12
NOW

www.hitex.de



Flex-BDM/68HC12
NOW

www.noral.com



CodeWarrior
NOW

www.hiware.ch



TRACE32-ICD
NOW

www.lauterbach.com



CABLE12
NOW

www.pemicro.com



EW6812
NOW

www.iar.com



EMUL12-PC-BDM
NOW

www.nohau.com

HCS12 Development Tools

Emulator



DProbeHC12
NOW

www.hitex.de



TRACE32-FIRE
NOW

www.lauterbach.com



ActivePOD
NOW

www.isystem.com



EMUL12-PC
NOW

www.nohau.com



JProbeHC12
NOW

www.hitex.de



EMUL12-PC-BDM
NOW

www.nohau.com



ActivePOD
NOW

www.isystem.com



Flex-BDM/68HC12
NOW

www.noral.com



TRACE32-ICD
NOW

www.lauterbach.com



CABLE12
NOW

www.pemicro.com

HCS12 Development Tools

- **Evaluation Boards**

- **HCS12 EVB**

- **Features:**

- Header footprints for access to all MCU pins
 - Headers for jumper selection of and connection to hardware options
 - Prototype expansion area for customized interfacing with the MCU
 - One CAN interface/driver
 - BDM IN and BDM OUT connectors for remote debugging of user's target system
 - RS-232C interface
 - 16 MHz crystal for 8-MHz operation
 - Single-supply +12VDC power input



- **Kit includes:**

- Evaluation Board (EVB)
 - CodeWarrior demo (IDE, Compiler, Debugger, Simulator)
 - Universal power supply
 - Serial Debug Interface (SDI) cable
 - Serial cable
 - CD ROM (user documentation)

- **Status: Available NOW**

CodeWarrior for Embedded Systems

CodeWarrior for HCS12

◆ Features:

• IDE

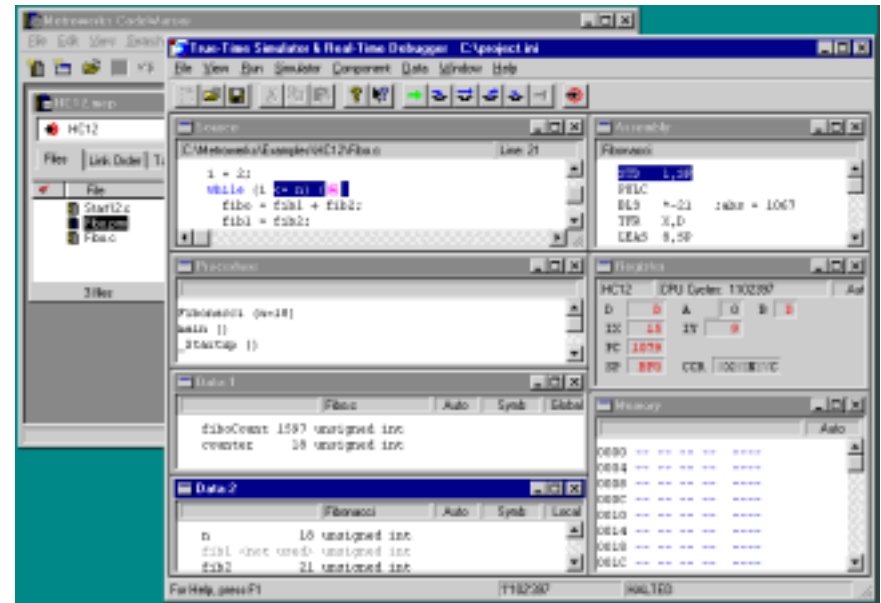
- Source-code editor
- Project manager
- Source code and symbol browser

• Build System

- Optimizing C, C++, EC++ compilers
- Assembler
- Linker dead strips unused code

• True Time Simulator

- Fast simulation (> 1.6 Mcycles/s)
- Fully configurable memory simulation
- I/O register awareness
- Event and interrupt handling for cycle accurate I/O simulation
- Time measuring, coverage and profiling analysis



• Debugger

- Source debugging in assembly, C and C++
- ORTI (OSEK Runtime Interface) for kernel aware debugging
- Flash programming utility
- Command line support

◆ Status: Available NOW



- Compiler/Debugger/Simulator

- cx6812 Compiler

- **Features:**

- IDEA - Windows 32-bit IDE
 - ANSI and ISO C Compiler
 - Reentrant and recursive
 - Position independent code
 - In-line assembly

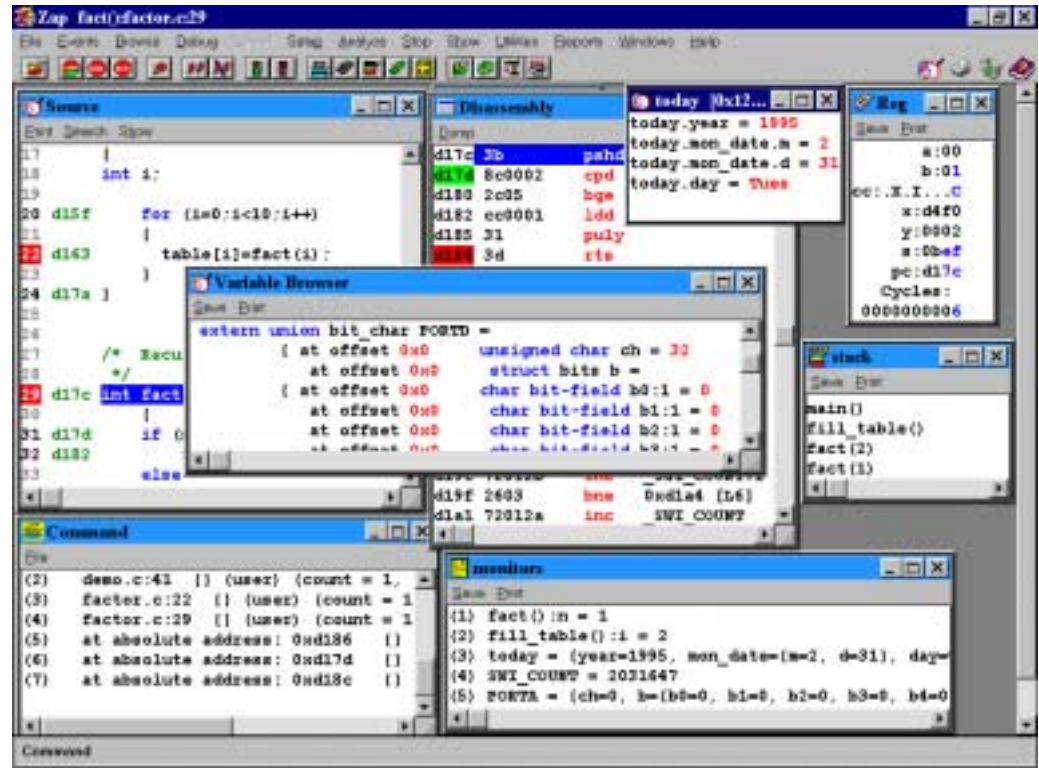
- **Status:** Available NOW

- ZAP 6812 Debugger

- **Features:**

- Nonintrusive "optimizer on" real-time debugging
 - Automated testing
 - Flash programming
 - Hardware breakpoints

- **Status:** Available NOW



- Embedded Workbench EW6812

- C Compiler

- **Features:**

- Highly efficient and PROMable code.
 - Full ANSI C compatibility.
 - Built-in chip-specific size and speed optimizer.
 - Target specific extensions.
 - Choice of memory models.

- **Status:** Available NOW

- C-SPY Simulator debugger

- **Features:**

- Supports all levels of debugging with complex breakpoints.
 - Debugging on C-source and assembly source levels.
 - Profiler for bottleneck analysis.
 - Code coverage.

- **Status:** Available NOW



- BDM Debugger/In-circuit Emulator

- JProbeHC12

- **Features:**

- Hardware and software breakpoints
 - Bank sensitive breakpoints
 - “Hot insert ” capability
 - Integrated flash and EEPROM programming

- **Status:** Available NOW

- DProbeHC12

- **Features:**

- Real-time emulation up to max.controller frequency
 - 1 MB dual ported emulation memory
 - 1 MB hard are breakpoints for code and data excellent adaptation solutions

- **Status:** Available NOW

- DProbeHC12 with Dtrace - **Trace extension with 32k frames**

- DProbeHC12 with Dbox

- **Features:**

- 64k trace frames (96 bits) with timestamp
 - Extensive trace filter options
 - Trigger system with sequence
 - Coverage and performance analysis





- In-circuit Emulator
 - ActivePOD - Emulator

- **Features:**

- Sandwich SMD technology
- High speed in-circuit emulation
- Real-time emulation up to 25MHz
- On board overlay RAM
- Integrated trace
- RAM-Based VLSI-FPGA design
- Reusable modules

- **Status: Available NOW**

- ActivePOD - Trace

- **Features:**

- 32/128k*160Bit trace buffer
 - Address (32)
 - Data (32/64)
 - Time Stamp (32)
 - Status (16)
 - AUX Signal (48/16)

- 100M samples/sec
- Programmable trigger unit
- High-speed trace RAM

- **Status: Available NOW**



- BDM Debugger/In-circuit Emulator

- TRACE32-ICD

- **Features**

- Cost effective debugger for C, C++ and java
 - Real time trace
 - High speed download up to 500 kbytes
 - Hardware breakpoints and on-chip trace
 - Flash-programming

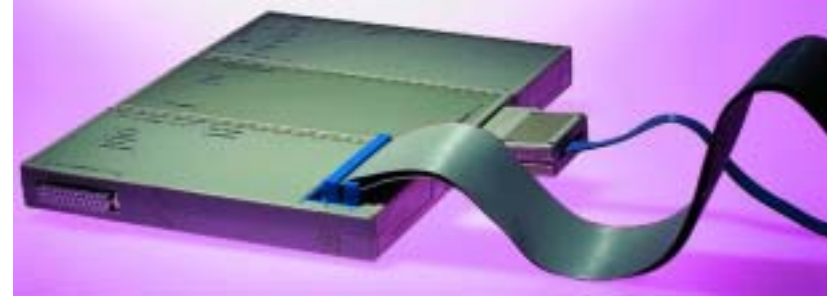
- **Status: Available NOW**

- TRACE32-FIRE

- **Features**

- First universal RISC emulator
 - 100 mhz+high speed emulation
 - High speed dual port emulation
 - Memory up to 8 MB
 - Bus and program flow trace
 - Performance-analysis, runtime statistics and code coverage analysis

- **Status: Available NOW**



- BDM Debugger/In-circuit Emulator

- EMUL12-PC/BDM

- **Features:**

- Supports HC12 BDM communications up to the maximum controller clock rate.
 - High Level Language (HLL) support for C compilers.
 - Shadow RAM for real-time viewing.
 - FLASH programmer.
 - Easy upgrade path to full Nohau emulators.

- **Status: Available NOW**

- EMUL12-PC

- **Features:**

- Full feature ICE. 12, 16 or 25 MHz depending on controller.
 - Compact hand held design goes anywhere.
 - Pipeline is fully decoded.
 - Unlimited no-skid hardware and software breakpoints.
 - Buy an ICE, get a BDM free.

- **Status: Available NOW**



- BDM Debugger
 - FLEX-BDM/68HC12 Debugger
 - **Features:**
 - Read and write target memory and CPU registers
 - Dynamic target access
 - Real-time hardware breakpoints
 - Unlimited execution breakpoints
 - Execute, break and single step control including 'go to function', 'go to module' etc.
 - Hot insertion (connect BDM while target executes) supported
 - Low power mode supported
 - Very low target clock speeds supported
 - Automatic optimization of BDM communications
 - High speed bi-directional parallel communications interface to host PC
 - **Status:** Available NOW



- BDM Debugger

- CABLE12 BDM Interface Cable/STAR Interface Cable

- **Features:**

- Full control of the on-chip resources via BDM
 - Full speed in-circuit emulation at a reasonable price
 - Programming for internal or external EEPROM/FLASH memory devices
 - External hardware breakpoint (optional)
 - Pullup options
 - Works with P&E's ICD12 debugger and PROG12 programmer

- **Status:**

- CABLE12 communicates with targets that run at up to 10 MHz internally and operate at 5 Volts. **Available NOW**
 - STAR Interface Cable communicate with targets that run at up to 32MHz internally and operate at 2.5 - 5 Volts. **Available NOW**

