

Die Motorola HC12 Mikrocontrollerfamilie

Lukas Fedorowicz

- Geschichtlicher Überblick Motorola
- Aufbau eines Mikroprozessors


- 68HC12 A-D Typen im Überblick
- Das Programmiermodell
- Interruptverarbeitung
- Betriebsarten
- Background Debug Mode

- Literaturverzeichnis

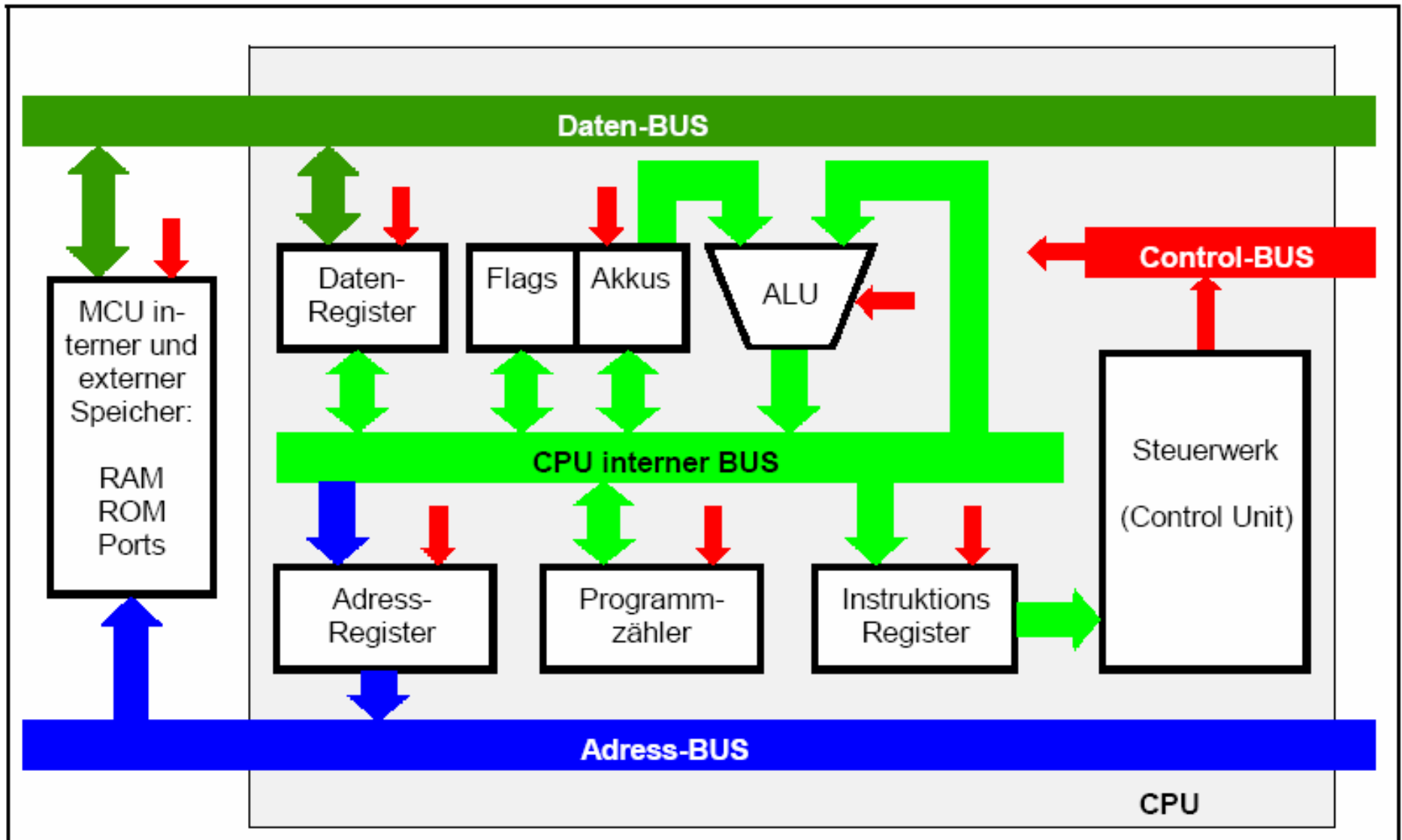


Die Motorola HC12 Mikrocontrollerfamilie Motorolas Geschichte

- 1928 von Paul und Joseph Galvin gründet
 - Manufacturing Corporation
 - 1947 in **Motorola** umbenannt
- Wortschöpfung aus Motor (motorcar, motion) und ola (Schall, Welle, la ola)
- 1930 erstes kommerzielles Autoradio
- 1969 Übertragung von Neil Armstrongs Worten zur Erde
- Schwerpunkte: Mobile Kommunikation, Halbleiter, Netzwerke und Embedded Systeme
- Bis 1985 Vorreiter auf dem Markt der Prozessoren
- Verbaut u. a. im Apple, Commodore Amiga, Atari ST, Sega Mega Drive oder NeoGeo



Mikroprozessor	Mikrocontroller
- nur Prozessor	- Prozessor + Peripherie
- in PCs, Workstations	- in techn. Anlagen
- Datenverarbeitung	- Steuerungen + Regelungen
- grosse Datenmengen	- mittlere Datenmengen
- langsame Dialogführung	- Echtzeitverarbeitung
- Größe eher unwichtig	- möglichst klein



- Steuerwerk aus Befehlsregister und Befehlsdecoder
 - Programmzähler
 - Adresse der nächsten Instruktion
 - Instruktionsregister
 - enthält den Code, der aktuell ausgeführt werden soll
 - Programmverarbeitung nach Von-Neumann-Zyklus

- Rechenwerk
- ALU
 - Zuständig für arithmetische und logische Funktionen
 - verknüpft die Werte der Operandenregister nach den Vorgaben des Steuerwerkes
- Hilfs- und Statusregistern
 - Zwischenspeicher für die Ergebnisse der ALU

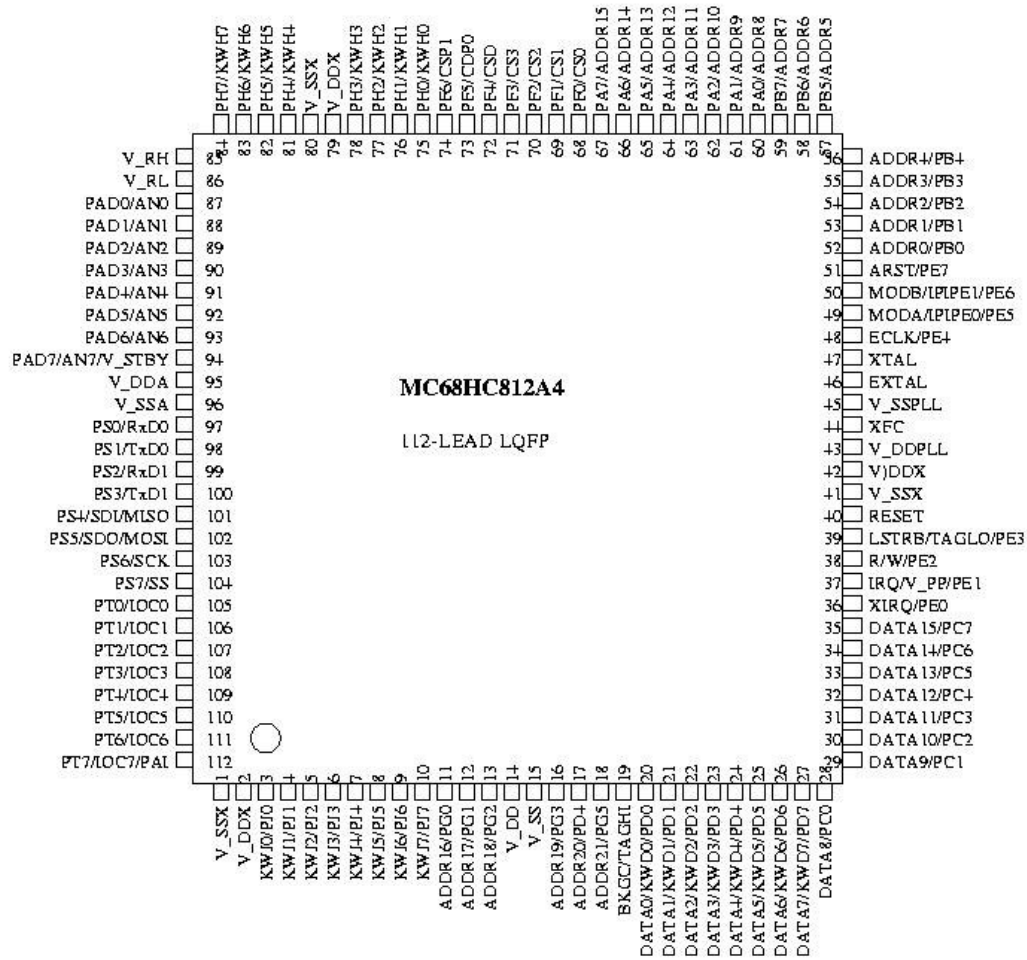
- Registerblock
 - Steuerung des Programmablaufs
 - Steuerung der Befehlsausführung
- Zusätzlich
 - Caches zur Erhöhung der Rechengeschwindigkeit
 - Fließkommarechenwerk
 - Pipelines
 - Interruptfähigkeit
 - Unterbrechung des Programmablaufes durch ein Signal.

Die Motorola HC12 Mikrocontrollerfamilie HCS11 - Vorgänger

- Vorgänger 68HC11
 - 8-Bit Von-Neumann-Architektur
 - breite Auswahl an Fachliteratur

- 68HC12 softwarekompatibel zu 68HC11
 - Struktur der CPU Register
 - Assemblerbefehle
 - Adressierungsarten





- HCS12 Familie
 - 16 Bit Prozessor
 - OnChip Peripherien
 - serielle Schnittstellen (SCI, SPI, IIC, CAN),
 - Timersystem mit Capture und Compare Einheiten
 - Pulsakkumulator
 - A/D Wandler
 - div. Überwachungsfunktionen wie Clock Monitor, Watchdog und Key Wakeup

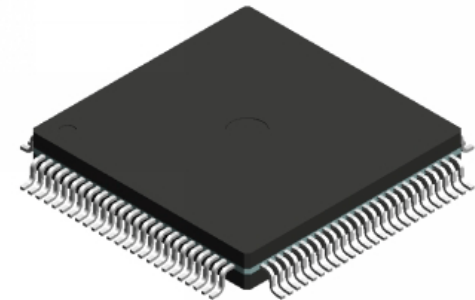
 - große Auswahl an Softwaretools
 - geringe Kosten

- MC68HC812A0
 - nicht mehr lieferbar ist.

- MC68HC812A4 – Urvater der HC12 Familie
 - 1K integrierten Speicher RAM
 - 4K EEPROM
 - CPU mit 16 Bit-Zentraleinheit
 - Timermodul mit Capture/Compare
 - zur Feststellung bzw. Erzeugung von zeitlich definierten Funktionen
 - synchrones serielles Interface (SPI)
 - von Motorola entwickeltes Bus-System mit dem digitale Schaltkreise miteinander verbunden werden können
 - 8-Bit A/D-Wandler mit 8 Kanälen
 - zwei asynchrone serielle Schnittstellen SCI0 und SCI1
 - über diese können Programme geladen werden

- Pulsakkumulator
 - Modul mit 8 Kanälen und 16 Bit Zählern, mit dem Ereignisse gezählt werden können.
- Betriebsspannung $5\text{ V} \pm 5\%$
 - muss gleichmäßig auf alle Versorgungsspannungspins verteilt werden.

- Ebenfalls CISC – Mikrocontroller
- 32K Flash-Memory OnChip
 - 10 Jahren Datenerhaltszeit
- Background Debug Mode (BDM) Interfa
 - Hardware Breakpoints
- A/D-Wandler
 - Auflösung auf 10 Bit erhöht
- Umgebungstemperatur zwischen -50 und 150 °C
- Gehäuse verfügt über 80 Pin im QFP Design
- Multiplexes Businterface
- Byte Data Link Communication BDLC Interface mit implementierten SAE Standard J1850

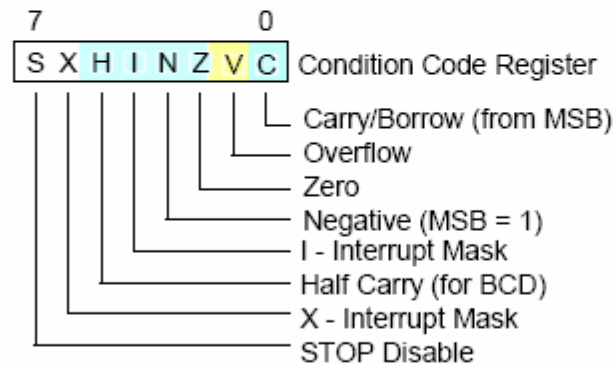


- B32 und BC32 sind pincompatibel
- msCAN12 – Modul statt BDLC
 - Reaktion auf die Anforderungen des europäischen Marktes
 - Controller Area Network – „Netzwerk aus mehreren Mikrocontrollern“

- 2 CAN Module beim 912DG128
- 3 CAN Module beim 912DT128
- Flashspeicher mit 60 KB bzw. 128 KB
- zweite SCI Schnittstelle
- Phase Locked Loop (PLL) zur Takterzeugung
- Zwei A/D Wandler
 - insgesamt 16 Analogkanäle (im TQFP112 Gehäuse)

68HC11 = 68HC12 = HCS12 Programmer's Model

7	A	0	7	B	0	8-bit Accumulators A and B
15	D				0	16-bit Double Accumulator D
15	X				0	Index Register X
15	Y				0	Index Register Y
15	SP				0	Stack Pointer
15	PC				0	Program Counter



HC05 / HC08 / HCS12

HC08 / HCS12

- Akkumulator-Architektur
 - Registermodell vom HC11 übernommen
 - 8-Bit Register A und B
 - Doppelregister D, aus A und B bestehend (das gleiche!)
 - Indexregister X und Y
- Stackpointer
 - Variablenübergabe beim Aufruf von Unterprogrammen
 - FIFO Prinzip
- Programmzähler
 - enthält die Adresse der nächsten auszuführenden Instruktion
 - Automatische Inkrementierung
- Statusregister

- Interrupts = Ereignisse, die die CPU veranlassen die aktuelle Codeausführung zu unterbrechen, den Inhalt sämtlicher CPU-Register zu sichern und zur Interruptserviceroutine zu springen.


Interrupt tritt auf:

1. Sicherung der CPU-Register + Programmzähler
2. Lookup in der Interrupt-Vektortabelle an \$FFFE:\$FFFF
 - 16 Bit Einträge
3. Ausführung der Subroutine
4. Rückkehr zum Hauptprogramm
 - RTI-Anweisung.
5. CPU Register restauriert
6. Fortführung des Hauptprogramms

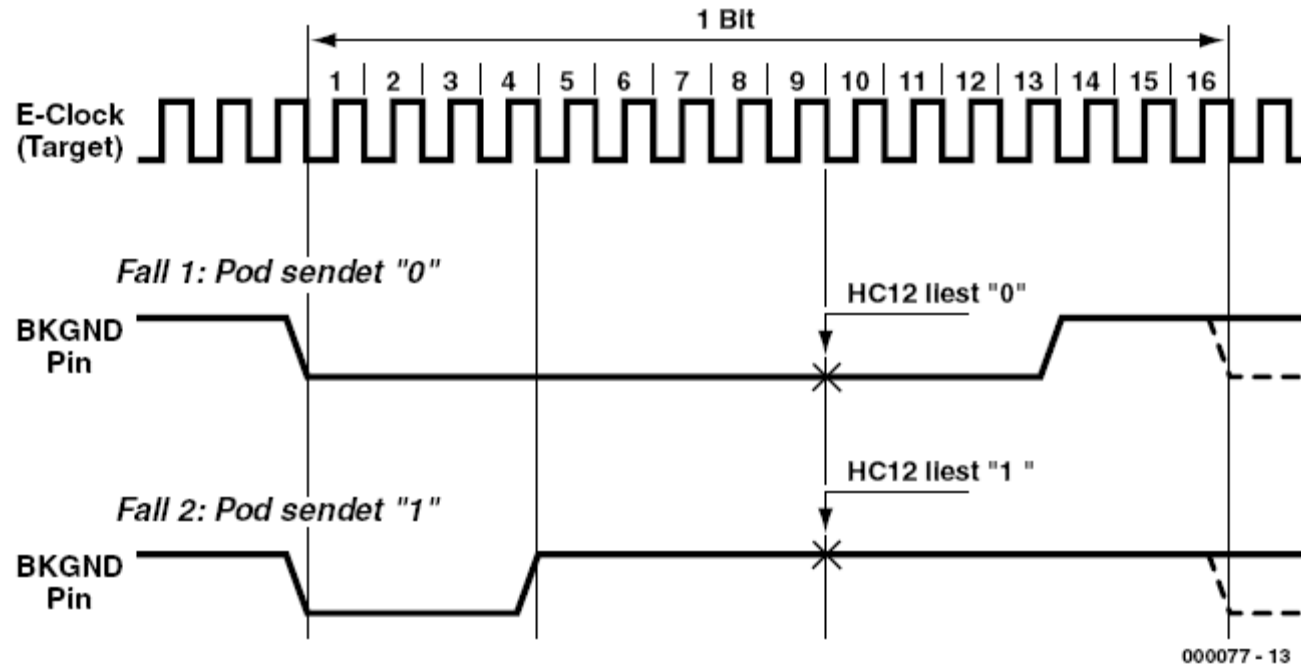
- Bei mehreren Interrupts
 - Reihenfolge in der Vektorabelle legt die Priorität fest
 - je höher die Adresse, desto höher ist die Priorität.
- Interrupt Latency
 - Taktzyklen im Worst Case vom Auftreten eines Interrupts bis zur Ausführung einer Interruptserviceroutine
 - wichtig für Echtzeitanforderungen
 - Hier: 21 Taktzyklen
 - längster Befehl (EMACS mit 13 Taktzyklen)
 - Zeit um die CPU Register zu retten (4 Taktzyklen)
 - Dauer um den Interruptvektor zu holen (1 Taktzyklus)
 - Füllen der Instruktionspipeline (3 Taktzyklen)
- Bei einer Busfrequenz von 8 MHz beträgt die Zykluszeit 40ns und somit die Interruptsverzögerungszeit 0,840µs.

- **Acht Modes**
 - nach dem Start oder Reset an MODA, MODB und MODC
 - MODE – Register
- **Single-Chip-Modus**
- **Expanded Modus**
 - externer Speicher kann adressiert werden
 - A und B Ports als Daten- und Adressbusse.
 - narrow
 - wide

- 4 Spezialbetriebsarten
 - für Testzwecke beim Hersteller Motorola reserviert
 - BDM nach dem Reset automatisch aktiv
- Peripheral Mode
 - Modus schaltet die CPU aus
 - externer Master muss die Kontrolle über die Peripheriebausteine übernehmen

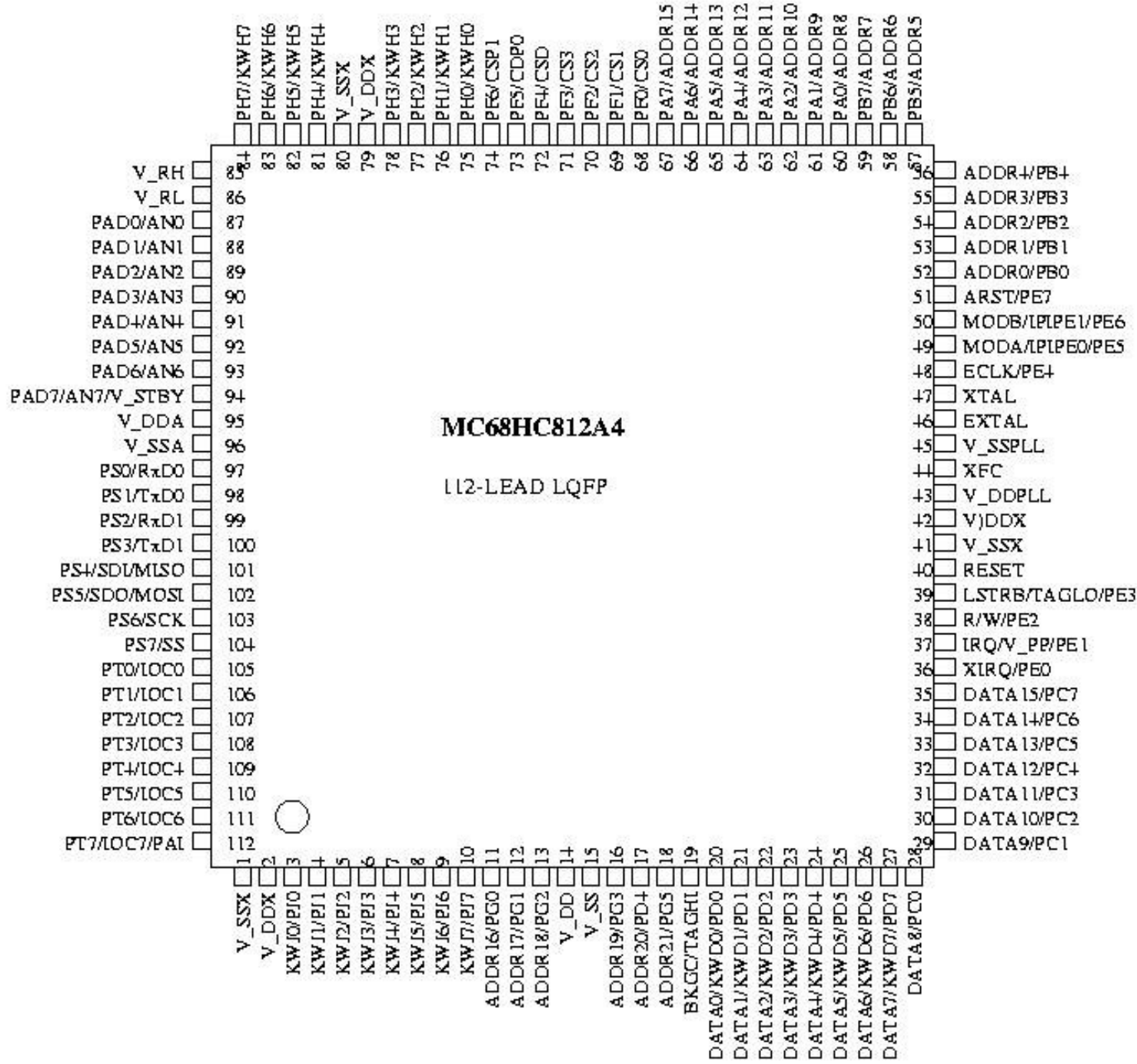
- 
-
- **Single-Wire Background Debug Mode**
 - Ein einziger Anschlusspin BKGD
 - Kein zusätzlicher Ressourcenverbrauch
 - funktioniert wie eine bidirektionale serielle Schnittstelle
 - eigenes Protokoll
 - Debugging im Hintergrund
 - Speicherzellen auslesen, ohne CPU unterbrechen zu müssen
 - Die BDM Befehle
 - Kommandobyte + erforderliche Parameter

- 16-MHz-Quarz -> E-Clock von 8 MHz
 - Basiszeiteinheit des BDM-Systems von 125 ns
- Übertragung eines einzelnen Bits sind 16 Taktzyklen erforderlich.
 - Bei 125 ns pro Takt werden somit $2 \mu\text{s}$ ($16 * 125\text{ns}$) für ein Bit benötigt
 - ein Byte $16 \mu\text{s}$.
- In der Praxis bedeutet das ca. 5-10 Tausend Befehle pro Sekunde.



- Host gibt ein Startimpuls vor
 - BKGD Ruhepegel ist ein HIGH ist
 - Synchronisation wird mit einem L-Pegel
- Übertragung einer logischen „1“
 - nach dem Synchronisationsimpuls
 - Host setzt die BKGD Leitung auf HIGH setzen.
 - 4 Takte später
 - BKGD tastet die Leitung ab -> erkennt den HIGH Pegel.
- Die Übertragung einer logischen „0“
 - Synchronisationsimpuls sendet Host
 - 13 Takte lang stabil auf LOW halten
 - drei Takte zurück auf HIGH.
 - MCU tastet vier Takte nach dem Erkennen der Synchronisation die Leitung ab
 - erkennt das Null-Bit.

- Pause zwischen zwei Bits < 512 Takte
- Sonst: Kommando wird verworfen
 - BDM wartet auf ein neues Befehl



- **Literaturhinweise**

- Mikrocontroller-Design, Harald Kreidl, 2003 Carl Hanser Verlag
- Programming the Motorola M68HC12 Family, Gordon Doughman:
- [http://dg3aaf.no-ip.com:8080/elektor/16-Bit-Mikrocontroller%20HC12%20\(I\).pdf](http://dg3aaf.no-ip.com:8080/elektor/16-Bit-Mikrocontroller%20HC12%20(I).pdf)
- <http://www.motorola.com/content.jsp?globalObjectId=1223-2212>
- <http://de.wikipedia.org/wiki/Motorola>
- <http://elmicro.com/hc12web/du9910/index.html>
- <http://www.seattlerobotics.org/encoder/apr98/68hc912intro.html>
- http://elmicro.com/files/intro_mc68hc12.pdf
- <http://hc12web.de>
- http://www.physics.ubc.ca/~scho/9s12/HCS12_sjtu/HCS12_Overview.pdf

Fragen ?