

Christian Streicher

Schneller und komfortabler

Konzeptstudie über die CMOS-CPU R65C02

Mit dem Erscheinen dieses Aufsatzes sind von Rockwell die ersten Muster der in CMOS-Technologie aufgebauten 6502-CPU verfügbar. Aus diesem Anlaß soll in der folgenden Betrachtung auf die Unterschiede in Funktion und Befehlssatz gegenüber der herkömmlichen NMOS-Version eingegangen werden.

Bedingt durch die verwendete CMOS-Technologie (Complementary Metal Oxid Silicon) ergeben sich vor allem in der Stromaufnahme erhebliche Verbesserungen gegenüber der NMOS-CPU (vgl. mc 1/1983, Seite 70). Vergleicht man die Leistungsparameter der für den kommerziellen Temperaturbereich spezifizierten NMOS-Version (575 mW) mit denen der R65C02-CPU (20 mW bei 1 MHz), so ergibt sich eine Reduzierung um 96,5 %. Ferner ist bei der CMOS-CPU die Leistungsaufnahme mit 20 mW/MHz direkt proportional zur Taktfrequenz. Hält man den Takt an, geht der Prozessor in den Standby-Modus, wobei bei 5 V Versorgungsspannung lediglich 2 µA zur Versorgung der internen Register benötigt werden. Dies entspricht im Vergleich etwa dem Reststrom einer Datenleitung der NMOS-CPU im hochohmigen Zustand. Im Gegensatz zum NMOS-Typ, der mit 100 kHz unterer Grenzbetriebsfrequenz spezifiziert ist, ist es beim R65C02 möglich, den Takt bis hin zum rein statischen Signal zu reduzieren. Ermöglicht wird dies durch den Aufbau der internen Register, die ähnlich wie ein statisches RAM organisiert sind. Wird nun wieder ein Taktsignal angelegt, fährt die CPU mit der Operation fort, die zuletzt im Befehlsregister abgelegt wurde. Neben der Möglichkeit des Standby-Betriebes können hierdurch auch extrem schnelle und sichere Direktzugriffe in den Speicher (DMA) realisiert werden. Die Synchronisation der 2-, 3- und 4-MHz-Versionen mit langsamen Arbeitsspeichern erfolgt ausschließlich über den RDY-Eingang der CPU. Während dieser bei der NMOS-6502 einem sehr genauen Timing unter-

legen war und das Anhalten des Bus-transfers nur zu bestimmten Zeitpunkten gestattet, reagiert die CMOS-CPU unmittelbar auf das eintreffende Steuerungssignal. Geschieht dies während eines Lesezugriffes, wird die momentane Adresse auf den Adressen-Anschlüssen angezeigt. Während eines angehaltenen Schreibzugriffes stehen an den Pins

D0-D7 zudem noch die aktuellen Daten zur Verfügung. Die Leistungsaufnahme bei aktivem RDY-Signal ist auf 10 mW/MHz reduziert.

Ideal für Batteriebetrieb

Die Versorgungsspannungstoleranz beträgt bei der CMOS-CPU $\pm 20\%$ gegenüber $\pm 10\%$ bei der NMOS-Version. Die somit zulässigen Betriebsspannungen zwischen 4 und 6 V ermöglichen selbst in batteriebetriebenen Computersystemen noch einen sicheren Betrieb. Ein Vergleich der Störspannungsabstände fällt ebenfalls zugunsten der CMOS-Technologie aus. Während das Rauschen auf den Leitungen bei der herkömmlichen Version 0,4 V nicht überschreiten darf, ist die eindeutige Zuordnung von Logiksignalen bei einem Grundrauschen von bis zu 0,8 V beim R65C02 noch gewährleistet.

Das Rücksetzen der CPU beim Einschalten der Versorgungsspannung kann über eine RC-Logik am Reset-Eingang erfolgen. Die nötige Flankensteilheit des Signals wird über einen integrierten Schmitt-Trigger gewährleistet. Bislang waren hierfür Zeitglieder erforderlich,

```

0001          ; BISHIER ANGEWANDTE TECHNIK FUER DIE BERECHNUNG VON
0002          ; SPRUNGLEISTEN
0003          ;
0004 0000          ==#0
0005 0000          JMPI  ==#+2
0006          ;
0007 0002          ==#1000
0008          ;
0009 1000 1C 10    ZIELE  .WORD  ZIEL1, ZIEL2, ZIEL3
0010          ;
0011          ; DAS INDEX REG. -X ENTHAELT DIE NUMMER DES
0012          ; SPRUNGVEKTORS, HIER 0 2
0013          ;
0014 1006 8A          JUMP  TXA          ; VEKTOR MIT 22 MULTIPLIZIEREN
0015 1007 0A          ASL   A
0016 1008 AA          TAX
0017 1009 BD 00 10    LDA  ZIELE, X    ; UMKOPIEREN DES WERTES
0018 100C 85 00      STA  JMPI
0019 100E BD 01 10    LDA  ZIELE+1, X
0020 0 1011 85 01     STA  JMPI+1
0021 1013 6C 00 00    JMP   (JMPI) ; AUSFUEHRUNG DES SPRUNGES
0022          ; ##### 16 BYTES/36 ZYKLEN #####
0023          ;
0024          ;
0025          ; VERWENDUNG DES NEUEN SPRUNGBEFEHLS
0026 1016          .OPT  IVB          ; SELEKTIERUNG DER CMOS-BEFEHLE
0027 1016 8A          JMPI  TXA          ; MULTIPLIKATION MIT 2
0028 1017 0A          ASL   A
0029 1018 AA          TAX
0030 1019 7C 00 10    JMP   (ZIELE, X) ; AUSFUEHRUNG
0031          ; ##### 6 BYTES/12 ZYKLEN #####
0032          ;
0033 101C          ZIEL1  ==
0034 101C          ZIEL2  ==
0035 101C          ZIEL3  ==
0036          ;
0037          .END

```

Bild 1. Einsparung von Bytes und Programmlaufzeit durch den indizierten indirekten Sprungbefehl der CMOS-CPU R65C02

die den Reset-Impuls um mindestens 10 Taktzyklen verzögern mußten, um unter anderem die Synchronisation der internen Logik für die dynamischen Register sicherzustellen. Ferner wird durch die Verwendung des Schmitt-Triggers gewährleistet, daß alle Peripheriebausteine von der CPU rückgesetzt sind.

Weitere Verbesserungen wurden im besonderen im zeitlichen Ablauf der Bus-signale vorgenommen. Stellvertretend sei hier die Verzögerungszeit bis zum Anlegen einer gültigen Adresse genannt. Diese beträgt bei der CMOS-CPU bei 2 MHz Systemtakt maximal 100 ns, wogegen die NMOS-Version mit 140 ns spezifiziert ist. Daraus resultierend beträgt die Zugriffszeit vom Prozessor auf den Speicher während eines Lesezyklus 340 ns gegenüber 310 ns bei der NMOS-Version. Die Gatterlaufzeiten der Adreßdecoder sind daher auch bei höherer Taktfrequenz unkritischer.

Die Mikroprozessoren der R65-CMOS-Familie werden in drei verschiedenen Versionen erhältlich sein. Der R65C02 ist, mit Ausnahme der bisher erwähnten Differenzen, identisch mit der NMOS-Version. Der R65C102 verfügt zudem über einen internen Oszillator sowie einen Quadratur-Phasenausgang, der das Anliegen einer gültigen Adresse signalisiert. Für Multiprozessor-Applikationen ist der R65C112 vorgesehen, der über einen Zwei-Phasen-Takteingang mit der Haupt-CPU verknüpft wird. Die beiden zuletzt erwähnten Typen sind zudem mit speziellen Steuerleitungen zur DMA-Kontrolle ausgestattet. Über einen Bus-Enable-Eingang können so der Daten- und Adreßbus sowie die R/W-Leitung hochohmig vom Bus getrennt werden. Ein „Memory-Lock“-Ausgang kennzeichnet einen „Read-Modify-Write“-Zugriff (bei DEC, INC, ROL, ROR, ASL, LSR), bei dem die CPU nicht unterbrochen werden darf.

Zahlreiche neue Befehle

Der Standardbefehlssatz der R65C02 ist mnemonisch und im erzeugten Operationscode kompatibel zur NMOS-Version. Dies gestattet es, alle existierenden Programme ohne Änderung auf den CMOS-Prozessor zu übernehmen. Dabei sollte jedoch unbedingt darauf geachtet werden, daß das ursprüngliche Programm ausschließlich Befehle enthält, die von den Herstellern spezifiziert sind. Eine Reihe von Operationscodes führen in der Tat auf dem NMOS-6502 sinnvolle Befehle aus, obwohl sie in keiner Be-

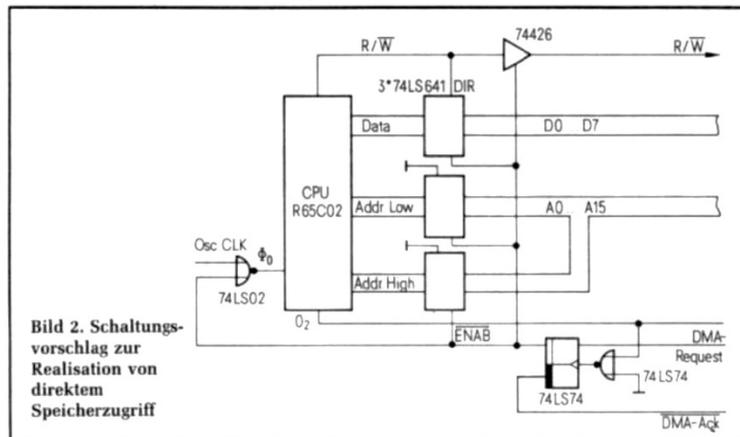


Bild 2. Schaltungsvorschlag zur Realisation von direktem Speicherzugriff

fehlliste zu finden sind. Der CMOS-Prozessor dagegen behandelt alle nicht implementierten Bitkombinationen als „No Operation“ mit unterschiedlicher Befehlslänge. Ferner wurde die Abarbeitung eines indirekten Sprungs (6C) dahingehend verbessert, daß dessen Operand nun auch über einer Seitengrenze stehen darf.

Die Befehlssatzerweiterung des R65C02 gliedert sich in drei Hauptgruppen:

1. Befehle mit neuen Mnemonics
 2. Neue Adressierungsarten
 3. Bit-Manipulationsbefehle
- Die neuen Mnemonics sind wie folgt gegliedert:

a) Programmkontrollbefehle

- BRA Branch always = unbedingter, relativer Sprung
- PHX Push X = Rette X-Register auf dem Stack
- PHY Push Y = Rette Y-Register auf dem Stack
- PLX Pull X = Laden des X-Registers vom Stack
- PLY Pull Y = Laden des Y-Registers vom Stack

b) Speichermodifizierende Befehle

- TSB Test and Set Bit = Vergleichen und Setzen der Bitmaske im Akkumulator (Adressierungsarten: Absolut/Zero Page)
- TRB Test and Reset Bit = Vergleichen und Rücksetzen der Bitmaske im Akkumulator (Adressierungsarten: Absolut/Zero Page)
- STZ Store Zero = Löschen einer Speicherzelle (Adressierungsarten: Absolut/Zero Page/Absolut, X/Zero Page, X)

Die neu hinzugefügten Adressierungsarten sind im 6502-Programmiermodell bereits enthalten. Es wurden lediglich die Zuordnungen für die verschiedenen Befehle erweitert.

So wurde den arithmetischen und logischen Befehlen die indirekte Adressierung über die Zero Page ohne Zuhilfenahme der Indexregister hinzugefügt. Diese, bisher nur bei JMP-Indirekt-Absolut gestattete Adressierung, unterstreicht vor allem die Programmierphilosophie der 6502-Familie. Da hier die gesamte Zero Page als Speicher für maximal 128 externe Indexregister benutzt werden kann, war bislang immer die Einschränkung gegeben, ein internes Indexregister für die entsprechenden Adressierungsarten abstellen zu müssen. Vor allem bei der in der Datenverwaltung häufig benutzten indexsequentiellen Speicherzugriffstechnik waren dadurch erhebliche Einschränkungen hinzunehmen.

Bei der Berechnung von Sprungtabellen waren bisher ebenfalls einschränkende Umladeprozesse erforderlich. Die Einführung eines absolut indiziert, indirekten Sprunges vereinfacht somit den Programmablauf (Bild 1) und reduziert die Rechenzeit.

Außerdem wurde der BIT-Befehl um die Adressierungsarten Immediate, Zero-Page-indiziert und absolut-indiziert erweitert. So wird ermöglicht, Speicherfelder mit bis zu 256 Elementen auf ein gesetztes Bit in einem Feldelement hin zu untersuchen, ohne den Akku vor jedem Test neu zu laden. Die Rechenzeitsparung beträgt bei einer Prüfung von 256 Bytes 512 Taktzyklen.

Tabelle: Assembler-Schreibweise der Bit-Manipulationsbefehle

Mnemonic	Bit-Nr.		Adresse		relative Distanz (BBR, BBS)
RMB	0-7	,	00-\$FF		
SMB	0-7	,	00-\$FF		
BBR	0-7	,	00-\$FF	,	00-\$FF (-128...+127)
BBS	0-7	,	00-\$FF	,	00-\$FF (-128...+127)

Schließlich wurden den INC- und DEC-Befehlen noch die Akkubezogene Adressierung hinzugefügt. Der dementsprechende 1-Byte-Befehl verändert den Akkumulatorinhalt in nur zwei Taktzyklen.

Bit-Manipulation

Die neu hinzugefügten Bit-Manipulationsbefehle, die ursprünglich für die kürzlich vorgestellten Einchipmikrocomputer (R6500/11) konzipiert wurden, weichen bei der Betrachtung der Standard-6502-Mnemonics vom gewohnten Aufbau ab. Diese basieren nämlich auf einem Operandenfeld, das bis zu 3 Datenelemente enthalten kann (Tabelle). Bisher waren lediglich eine Adresse und die Zuordnung eines Indexregisters üblich.

Bei der Assemblierung wird die Bit-Position in den Operationscode impliziert

(vgl. Befehlscode-Matrix), so daß die Bit-Set-/Reset-Befehle auf 2 Byte Befehlslänge reduziert werden. Die Sprungkommandos sind dementsprechend auf einen 3-Byte-Befehlscode aufgebaut. Bei manueller Korrektur bzw. Berechnung der Sprungdistanzen ist jedoch darauf zu achten, daß der Programmzählervorlauf 3 Byte und nicht, wie bei allen übrigen Branch-Anweisungen, 2 Bytes beträgt.

Applikations-Tips

Die wohl interessantesten Neuanwendungen des CMOS-6502 ergeben sich aus dem statischen Verhalten der internen Register. Stellvertretend für derartige Applikationen sei hier der Grundgedanke für den Aufbau einer schnellen DMA-Steuerung behandelt. Da der R65C02 noch nicht über den bereits er-

wähnten Bus-Enable-Eingang verfügt, müssen bidirektionale Treiber mit Enable-Eingang unmittelbar an die Ausgänge der CPU geschaltet werden. In der Regel läßt sich dies aufgrund der starken kapazitiven Busbelastung der RAMs ohnehin nicht umgehen. Die Enable-Eingänge der Treiber für den Daten- und Adreßbus sowie die R/W-Steuerleitung dienen nun als DMA-Request-Eingang.

Für diesen Eingang sei negative Logik vorausgesetzt (z. B. 74LS644). Das DMA-Request-Signal schaltet so über den logischen Wert 1 alle Treiber in den hochohmigen Zustand. Zudem wird der Ausgang des Taktgenerators mit dem DMA-Signal logisch NOR-verknüpft. Der Ausgang dieses Schaltkreises liegt am $\Phi 0$ -Eingang des R65C02. Dies hat zur Folge, daß mit dem Auslösen des DMA-Request der Takt der CPU auf den logischen Wert 0 geschaltet wird. Das DMA-Signal sollte aus Sicherheitsgründen nur mit der fallenden Flanke von $\Phi 2$ ausgelöst werden. Die Synchronisation kann durch ein D-Flipflop geschehen; sein Ausgangssignal des FF kann zudem als DMA-Acknowledge dienen. Durch das Anhalten des Taktes kann so ein DMA-Zugriff ohne Datenverluste mit einer Reaktionszeit von maximal einer Taktphase ausgelöst werden. Das Gesamtkonzept der Schaltung ist in Bild 2 dargestellt.

Die Einsparung an Rechenzeit und Speicherplatz durch Verwendung der neuen Befehle lassen sich am eindrucksvollsten am Beispiel der Registerabspeicherung auf den Stack demonstrieren. Die Aufgabenstellung sei, die internen Register X und Y auf dem Stack zu hinterlegen, ohne den Akkuinhalt zu zerstören. Dieses Kriterium wird meist innerhalb von System- oder Monitorprogramm gefordert, um innerhalb der Ein-/Ausgeroutine die Datenvektoren nicht zu zerstören. Bild 3 zeigt die Gegenüberstellung beider Lösungsmöglichkeiten. Zur Entwicklung des R65C02 stehen derzeit ein Makro-Assembler mit Linking-Loader für das System 65 von Rockwell zur Verfügung. Die neuen Befehle können jedoch auf jedem 6502-Assembler durch BYT-Statements implementiert werden. Ein In-Circuit-Emulator mit bis zu 4 MHz Taktgeschwindigkeit ist ab Mitte dieses Jahres für das Rockwell-Development-Center (RDC) verfügbar.

```

0001 0000          **$1000
0002          ; RETTEN VON X & Y AUF DEM STACK (NMOS-VERSION)
0003          ;
0004          ; DARF NUR ALS UNTERPROGRAMM BENUTZT WERDEN !!
0005          ;
0006 1000 48      SAVXY PHA          ; ZWEI DUMMIES AUF DEN STACK
0007 1001 48      PHA
0008 1002 08      PHP          ; STATUS RETTEN
0009 1003 48      PHA          ; ACCUM. RETTEN
0010 1004 8A      TXA
0011 1005 48      PHA          ; X-REG. ZWISCHENSPEICHERN
0012 1006 BA      TSX          ; STACK-INDEX HOLEN
0013 1007 BD 07 01 LDA          $107, X ; RUECKSPRUNGADRESSE HOLEN
0014 100A 9D 05 01 STA          $105, X
0015 100D BD 06 01 LDA          $106, X
0016 1010 9D 04 01 STA          $104, X
0017 1013 98      TYA
0018 1014 9D 06 01 STA          $106, X ; Y-REG. AUF DEN STACK
0019 1017 68      PLA
0020 1018 9D 07 01 STA          $107, X ; X-REG. AUF DEN STACK
0021 101B AA      TAX          ; X-REG. WIEDERHERSTELLEN
0022 101C 68      PLA          ; ACCUM. WIEDERHERSTELLEN
0023 101D 28      PLP          ; STATUS WIEDERHERSTELLEN
0024 101E 60      RTS
;
; ***** 31 BYTES/71 ZYKLEN Z*****
0025          ;
0026          ;
0027          ;
0028          ; RETTEN VON X & Y AUF DEM STACK (CMOS-VERSION)
0029          ;
0030          ; DARF IM PROGRAMMABLAUF STEHEN
0031          ;
0032          ; .OPT   IVB
0033          ;
0034 101F 5A      PHY          ; Y-REG. RETTEN
0035 1020 DA      PHX          ; X-REG. RETTEN
0036          ; ***** 2 BYTES/6 ZYKLEN *****
0037          ;
0038          ; .END

```

Bild 3. Retten von X- und Y-Register auf den Stack: Alte und neue Programmversion

Der Autor ist Mitarbeiter der Firma Rockwell International in Martinsried bei München.