

Mikroprozessor CP3-F Befehlssatz mit Hintergrundaspekten

1960 waren druckende Rechner oft noch rein mechanisch und anzeigende Rechner waren diskret aus Einzeltransistoren aufgebaut, was sich in Baugröße und einer Masse von 13 kg störend bemerkbar machte. Versuche, durch Einsatz von TTL-Bausteinen Größe und Gewicht zu reduzieren, erbrachten nur einen Teilerfolg. Vielversprechender erschien der Einsatz von deutlich höher integrierten Schaltkreisen mit der damals neuen PMOS-Technologie. So wurden erste Rechnerchips entwickelt. Sehr bald zeigte sich, dass diese zu unflexibel waren. Bei jeder Änderung mussten neue Masken erstellt und neue Bausteine entwickelt und erprobt werden. (Programmierbare MOS-Bausteine gab es damals noch nicht.) So wurde schnell erkannt, dass Systeme bestehend aus (möglichst wenigen) Standard- und spezifischen Programmbausteinen ein Optimum zwischen kurzen Entwicklungszeiten und hoher Flexibilität boten. Da anzeigende und druckende Rechner einen erkennbar großen Markt boten, schienen 4-Bit-Prozessoren ein geeigneter Weg, den künftigen Bedarf zu decken. So entwickelte Rockwell das Mikroprozessorsystem PPS4, welches 1972 auf den Markt kam (weitere Informationen dazu siehe: https://en.wikichip.org/wiki/rockwell_international/pps-4). Als Hersteller von Büromaschinen – also auch Schreibmaschinen – beschloss die Firma Olympia, auf eine 8-Bit-Architektur zu setzen, da damit außer Ziffern auch ASCII- oder ISO-codierte Buchstaben verarbeitet werden konnten. Ziel war, einen Rechner mit nur 2 Bausteinen zu entwickeln, was dazu führte, dass in dem Standard-Baustein „RSE“ (Rechen-Speicher-Einheit CPU) auch ein für einfache Rechner ausreichendes 48 Byte großes Daten-RAM und 8 Peripheriekanäle mit enthalten waren. Als Programmspeicher diente ein Baustein mit 1 kByte ROM und 16 I/O-Kanälen. So konnte ein Rechner mit 8-stelliger gemultiplexer Anzeige und Tastatur einfach realisiert werden: 8 I/O-Leitungen dienten dabei der Ansteuerung der 8 Stellen und gleichzeitig als Abfrageleitungen der Tastatur. 8 weitere I/O-Leitungen lieferten die Ansteuerung für die 7 Segmente der Anzeige und den Dezimalpunkt. Wenn 4 Leitungen der RSE für die Rückmeldungen von der Tastatur genutzt wurden, blieben noch 4 weitere für Sonderzwecke verfügbar. Vorgestellt wurde das System auf der Electronic Nov. 1974 in München.

Für größere Systeme waren ROM, RAM und I/O je nach Bedarf mit zusätzlichen Bausteinen erweiterbar. Diese Anforderungen hatten direkte Auswirkungen auf die Architektur und den Befehlssatz, wobei die damals möglichen Strukturweiten und die aus heutiger Sicht relativ geringe Verarbeitungsgeschwindigkeit Strukturen mit wenigen Flächenbedarf und geringer Zyklenzahl für die Abarbeitung der Befehle erforderlich machte.

Um Chipfläche in der RSE für Adressregister und Treiber für Adressleitungen zu sparen, wurden die Adressregister (teilweise) in die externen Bausteine verlegt. Jeder Baustein enthält 4 Adressregister zu je 11 Bit, nämlich den Programmcounter Q und einen Stack mit 3 Rückkehradressen RA, RB und RZ für Unterprogrammaufrufe, wobei das Register RZ auch als Datenpointer z.B. für Tabellenverarbeitung genutzt werden kann. (Beachtet werden muss dabei, dass bei einem Unterprogrammprung der Inhalt dieses Registers überschrieben wird.) Mit den 11 Adressbits kann ein Bereich von 2 kByte adressiert werden. Zusätzlich hat die CPU noch 6 Adressausgänge für die Ausgabe einer „Moduladresse“. Bei Sprüngen werden die unteren 3 Adressbits (aus dem Befehlscode des Sprungbefehls) in die Bits 9 bis 11 des Q-Registers übernommen, die oberen 3 Adressbits enthalten die „Moduladresse“, womit ein Adressbereich von 16 kByte möglich ist (nachzulesen unter: Ing. (grad.) Jürgen Sorgenfrei „Der erste europäische Mikroprozessor“, Sonderausgabe „Elektronik“ Heft „Mikroprozessoren“ 1977).

Der CP3-F wurde von SGS ATES unter dem Namen M38 auf den Markt gebracht. SGS hat dazu auch ausführliche Beschreibungen des Systems (in englischer Sprache) erstellt.

Befehle der CP3-F

Arithmetische Befehle:

ABL Code HEX 0E: addiere binär Literal: das folgende Byte „Literal“ wird zum Akku addiert

ABR Code HEX Ax: addiere Registerinhalt zum Akku, x Wert 0 bis HEX B ist die Adresse des RAM-Registers in der CPU, x Wert HEX C bedeutet Adressierung über S- und T-Register, x Wert D ist wie C, aber autodecrement, Wert xE bedeutet autoincrement.

ADR Code HEX Ex: wie ABL, aber Dezimaladdition

VGL Code HEX 0F: Vergleich Akku mit Literal, gesetzt werden nur Flags

VLL Code HEX 1C: Verschiebe Akku logisch 1 Bit nach links

VTL Code HEX 1E: Verschiebe Akku um 1 Tetrade (4 Bit) nach links

VLR Code HEX 1D: Verschiebe Akku logisch 1 Bit nach rechts

VTR Code HEX 1F: Verschiebe Akku um 1 Tetrade (4 Bit) nach rechts

Logische Befehle:

KOL Code HEX 05: Konjunktion (UND-Verknüpfung) Akku mit Literal

KOR Code HEX Bx: Konjunktion Akku mit Register, Adressierung wie bei ABR

EOL Code HEX 0C: Exklusiv-Oder Akku mit Literal

EOR Code HEX Cx: Exklusiv-Oder Akku mit Register, Adressierung wie bei ABR

ORL Code HEX 0D: Oder Akku mit Literal

Einen Befehl für Oder Akku mit Register gibt es nicht, kann durch Invertieren (mit EOL HEX FF), UND-Verknüpfung und erneutes Invertieren realisiert werden. Wird normalerweise selten gebraucht.

Laden und Speichern:

LAL Code HEX 04: Lade Literal (das folgende Byte) in Akku (2 Byte Befehl)

LAK Code HEX Fx: Lade Akku „kurz“ mit Wert x des Befehlscodes (1 Byte Befehl)

LAR Code HEX 8x: Lade Akku mit Registerinhalt, Adressierung wie bei Befehl ABR

LSK Code HEX 28 bis 2F: die unteren 3 Bit des Befehls werden in das Register „S“ geladen. Dieses dient mit dem „T“ Register zur Adressierung des internen RAMs, auch der auf andere Weise adressierbaren RAM-Zellen.

LTK Code HEX 28 bis 2F: die unteren 3 Bit des Befehls werden in das Register „T“ geladen.

LAM Code HEX 20 bis 27: die unteren 3 Bit des Befehls enthalten die Adresse des I/O-Ports, dessen Inhalt in den Akku geladen wird.

SAT Code HEX 01: die unteren 3 Bits des Akkus werden in das Register „T“ geladen

SST Code HEX 03: die unteren 6 Bits des Akkus werden in die Register „S“ und „T“ geladen

SAR Code HEX 9x: der Akku wird in das Register geladen, Adressierung wie bei Befehl ABR

SAM Code HEX 30 bis 37: die unteren 3 Bit des Befehls enthalten die Adresse des I/O-Ports, auf dem der Inhalt des Akkus ausgegeben wird.

Sprungbefehle

Bei allen Sprungbefehlen sind in den unteren 3 Bit des Befehlscodes die oberen 3 Bits der Adresse enthalten. Das folgende Byte enthält die unteren 8 Bit der Adresse. Damit kann jeder Sprung (bedingt oder absolut) den vollen 2 kByte Bereich des Programcounters erreichen.

SPR Code HEX 40 bis 47: absoluter Sprung

SGN Code HEX 48 bis 4F: springe wenn Akku Null

SUN Code HEX 50 bis 57: springe wenn Akku ungleich Null

SBP Code HEX 58 bis 5F: springe wenn Akku positiv

SSP Code HEX 60 bis 67: springe wenn Register „S“ ungleich 7 (Schleifenzähler ähnlich „DJNZ“ bei 8048 Prozessor, betrifft nur den Adresspointer in Verbindung mit autoincrement / -decrement)

SEU Code HEX 68 bis 6F: springe wenn Übertrag gesetzt ist

SKU Code HEX 70 bis 77: springe wenn kein Übertrag gesetzt ist

SUP Code HEX 78 bis 7F: Sprung in Unterprogramm

RSP Code HEX 00: Rückkehr aus Unterprogramm