


Cora, der erste Schweizer Transistorrechner

Report

Author(s):

Bruderer, Herbert 

Publication date:

2013

Permanent link:

<https://doi.org/10.3929/ethz-a-010001422>

Rights / license:

In Copyright - Non-Commercial Use Permitted

Cora, der erste Schweizer Transistorrechner

Der vom ungarischen Ingenieur Peter Toth entwickelte und gebaute erste Schweizer Transistorrechner Cora wurde 1963 fertig gestellt. Diese bei der Zürcher Contraves AG gefertigte Maschine war lange Zeit selbst in der Fachwelt kaum bekannt. 2011 wurde an der ETH Lausanne eine Cora 1 gefunden, sie steht heute im dortigen Bolo-Museum. Unseres Wissens hat nur eine einzige Cora 1 überlebt. Die Nachfolgemaschine Cora 2 entstand unter der Leitung des Schweizer Ingenieurs Peter Blum, sie wurde 1969 vollendet. Im Folgenden berichten zwei Fachleute, Ernst Hutzler und François Nicolet, über ihre Erlebnisse rund um die Cora.

Herbert Bruderer)*

Programmierung des Coragraphen

Rückblick auf meine Tätigkeit bei der Firma Contraves AG

Meine Anstellung als Programmierer bei der Firma Contraves AG, Zürich, dauerte von August 1968 bis Oktober 1974. Diesen Zeitraum will ich in drei Etappen nach den Namen der jeweils eingesetzten Rechner einteilen: Cora 1, Cora 2 und DEC-PDP 11. Cora steht für *Contraves Rechenanlage*.

Die näheren Gründe für eine Eigenentwicklung eines Rechners, die Firmenstruktur zivil/militärisch mit den beteiligten und federführenden Persönlichkeiten und die militärischen Einsatzgebiete bzw. die militärischen Spezialversionen sind mir nicht bekannt. Dieser Bericht bezieht sich nur auf meine Tätigkeit in der Software-Gruppe der (zivilen) Anwendung des Coragraphen (Zeichentisch). Eine erste, durch eine Cora 1 gesteuerte Zeichenanlage mit Bahnsteuerung wurde 1964 an der Expo 64, der Schweizerischen Landesausstellung in Lausanne, gezeigt.

Cora 1: von 1963 bis 1971

Beim Eintritt in die Contraves beschränkte sich meine Computererfahrung auf einige Algol-Programme, eingesetzt auf dem Rechner TR4 der Firma Telefunken. Von Maschinensprachen hatte ich keine Ahnung. In meiner ersten Woche erhielt ich zusammen mit Mitarbeitern des Kunden ETA (Uhrenfabrik) in Grenchen eine gründliche Assemblereinführung, die von Adrian Häne gehalten wurde. Bereits 1965 hatte er den bis 1971 gültigen Assembler mit den erforderlichen Dienstprogrammen programmiert. Wir alle, Klaus Kienke, Hansruedi Brunner, Arnold Vogt, Ruedi Leuthold, Dominic Gsell und einige andere, stützten uns auf seine Erfahrung und seine Grundlagen. Ausserdem entwickelte er das für den Datenaustausch unter den Anwendungsprogrammen wichtige *allgemeine Zeichenprogramm AZP*. Das Austauschformat, ebenfalls AZP genannt, wurde zu einem Quasi-Standardformat. Es wurde bis in die 1980er Jahre in Deutschland in verschiedenen Landesvermessungsämtern verwendet.

In den Jahren 1966 bis 1970 wurden Anwendungsprogramme für folgende Bereiche geschrieben:

- Maschinenbau,
- Katasterwesen,
- Bekleidungsindustrie (Schnittmuster),
- Strassenbau,
- Elektronik, gedruckte und integrierte Schaltungen,

- nautische und aeronautische Karten.

Die Programme mussten mit den 4096 24-Bit Worten (4 Kilobyte) des Hauptspeichers auskommen. Zumindest anfangs gab es keine Möglichkeit zum Auslagern von Daten und Programmteilen auf Magnetplatten und Magnetbänder. Die Daten auf Lochstreifen wurden von den Anwendungsprogrammen verarbeitet. Diese steuerten einen mit entsprechenden Werkzeugen ausgestatteten Zeichentisch. Bei Bedarf wurden begleitend eine Liste und ein Resultat in Form eines Lochstreifens erstellt. Dieses Ergebnis konnte wieder als Eingabe für eine Verarbeitung mit mehreren Durchläufen genutzt werden. Der erneut erzeugte Lochstreifen diente für den nächsten Durchgang usw.

Um die Anforderungen der Bahnsteuerung erfüllen zu können, war „ballastfreies“ Programmieren zwingend. Die Benutzerfreundlichkeit kam erst an zweiter Stelle. Es gab kein eigentliches Betriebssystem (Monoprogrammierung). Die Systembefehle wurden in das Anwendungsprogramm eingebettet. Unbenutzte Bits und Bytes von Befehlen ohne Adressen wurden als Zähler, Konstanten und Flags (Markierungen) verwendet. „GoTo“ war ein geeigneter Befehl. Die ein bis zwei für die Mitarbeiter bereit stehenden Rechner wurden tagsüber von der Applikationsgruppe zum Teil mit und für Kunden gebraucht. Deshalb waren die rechnerbedingten Arbeiten wie Assemblierung, Testen sowie Fehlersuche nachts durchzuführen. Das Ausdrucken einer Programmliste dauerte bis zu vier Stunden.

Die Suche nach Fehlern und die entsprechenden Korrekturen waren mit den Schaltern zu bewerkstelligen. Zuerst wurde die Adresse eingestellt, dann ein neuer Befehl gesetzt – ein Schalter pro Bit –, und dann wurde der Befehl in den Speicher übertragen. Erweiterungen mussten vorübergehend mit unbedingten Sprüngen in noch nicht belegte Bereiche ausgelagert werden, zusammen mit einem anschließenden Rücksprung an die richtige Adresse. Sobald eine gewisse Menge an Änderungen vorlag und solange das Programm noch funktionierte, wurde der Quellcode bearbeitet, mit einer neuen Programmliste assembliert, und das Testen konnte wieder beginnen. Planung und Programm-entwürfe (mit Schreibstift und Papier) wurden tagsüber vorgenommen.

Kenndaten der Cora 1 im Vergleich zu heute gebräuchlichen Tischrechnern

Hauptspeicher

damals	heute	Verhältnis
4096 Worte x 24 Bit (= 98 304 Bit)	4 Gigabyte x 8 Bit = 32 GBit (32 000 000 000 Bit)	1 : 300 000

Taktzeit

damals	heute	Verhältnis
6 Mikrosekunden	2 Nanosekunden	3 000 : 1

Preis (inflationbereinigt)

damals	heute	Verhältnis
150 000 SFr.	1 000 SFr.	150 : 1

Bis 1971 wurden zwar die Ein- und Ausgabegeräte sowie die Massenspeicher (Digitalisiergerät, Kurvenzeichner, Magnetbänder usw.) laufend zeitgemäss erneuert und erweitert, die Rechner wurden jedoch vom Militärbereich unverändert geliefert. Während wir 41 Systeme (Cora 1) verkaufen konnten, hatte die Konkurrenz in den USA, allein in den grafischen Anwendungen tausende Anlagen ausgeliefert. Contraves versuchte mit der Neuentwicklung Cora 2 am Markt zu bleiben, musste aber gegen sehr leistungsfähige Computer, z.B. PDP 11 – eine Reihe der Firma Digital Equipment Corporation, mit hunderten von Entwicklern antreten.

Die für den Coragraphen notwendige Elektronik, Mechanik und Software wurden im Industriebereich (Zivilbereich) unter der Leitung von Heinz König entwickelt. Insgesamt mit der Applikationsgruppe und dem Verkauf umfasste 1971 die gesamte Abteilung 40 Mitarbeiter.

Cora 2: von 1971 bis 1973

Wesentliche Verbesserungen gegenüber Cora 1 waren:

- ein Interrupt-System (Interrupt = Programmunterbrechung), welches einen Mehrprogrammbetrieb ermöglichte,
- ausbaubare Speicher in Pages (Speicherblöcke im virtuellen Speicher) zu 4 096 Worten,
- Wechselplatten,
- Magnetbandeinheiten.

Parallel zur Erhöhung der Leistungsfähigkeit des Rechnersystems wurden die Arbeitstische verbessert und die Bearbeitungswerkzeuge erweitert. Für die hochpräzise Fertigung von Filmen für integrierte Schaltkreise wurden sehr genaue Tische und Spezialwerkzeuge entwickelt. Sehr rasch konnten einige Cora-1-Anwendungen umgeschrieben werden. Als Schwachstelle zeigte sich ein ebenso rasch erstelltes Betriebssystem. Erst als Arnold Schneider ein umfangreiches Betriebssystem aufbaute, das auch für den zeitkritischen Betrieb von zwei Zeichentischen genügte, konnten die bisherigen Anwendungen erweitert werden. Das Betriebssystem belegte dabei die Basis-Page (Basisspeicher). Die Anwendungsprogramme beanspruchten einen zusätzlichen Speicherbereich, mindestens eine Page. Um den Rechner einem grösseren Benutzerkreis zugänglich zu machen, schrieb François Nicolet einen Fortran-IV-Compiler. Compiler waren damals eher auf den Grossrechnern anzutreffen. Für die doch immer noch recht kleine Speichergrösse der Cora 2 war es eine grosse Errungenschaft, eine höhere Programmiersprache anbieten zu können. Auf wie vielen der 18 ausgelieferten Cora 2 dieser Compiler eingesetzt wurde, ist nicht bekannt.

Jedoch bereits zwei Jahre nach der Einführung der Cora 2 wurde der Entscheid getroffen, auf einen Fremdrechner zu wechseln. Die von den Kunden geforderten Leistungen machten es unumgänglich. Seit 1971 bot die Firma Digital Equipment Corporation (DEC) die Rechnerserie PDP 11/45-70 an. Diese weit verbreiteten Rechner waren für unsere Bedürfnisse besser geeignet. Sie liessen sich über ihre Bussysteme gut mit Fremdprodukten verknüpfen, sie hatten ein beachtliches Preis-Leistungs-Verhältnis.

DEC PDP 11: ab 1973

Wieder mussten bestehende Anwendungsprogramme angepasst und umgeschrieben werden. Man war aber von den oben genannten Fesseln befreit. Die bisherige rein sequentielle Verarbeitung ging in die Handhabung mit interaktiven Systemen über. Direkt zugreifbare Daten auf Platten mit einer realistischen Darstellung auf dem Bildschirm (Wysiwyg, „what you see is what you get“, gleiche Anzeige am Bildschirm wie im Ausdruck) liessen völlig neue Anwendungskonzepte zu. Ein Entwurf für ein solches interaktives System LINGRAS war dann auch meine Abschlussarbeit.

Nach sechs interessanten, turbulenten und auch anstrengenden Jahren wechselte ich in das Institut für Kartographie der ETH Zürich und arbeitete dort mit den bei der Firma Contraves gemachten Erfahrungen weiter an interaktiven Systemen.

Ernst Hutzler

Häuslenen, 28.2.2013

Bau eines Fortran-IV-Compilers für die Cora 2

Der Physiker François Louis Nicolet aus Zürich arbeitete von Anfang Oktober 1971 bis Mitte Februar 1975 bei der Zürcher Firma Contraves. Sein Vorgesetzter war Heinz König, die Kollegen hiesigen Dominic Gsell, Adrian Häne, Hansruedi Brunner, Hansruedi Hürlimann, Heinz Busch (Kartograf) und Arnold Schneider. Sie haben entweder am Übersetzungsprogramm (Compiler) oder an Anwendungsprogrammen gearbeitet. Nach dem Studium war Nicolet zunächst am Cern in Genf tätig, wo er sich der Informatik zuwandte. Er schildert den Bau eines Fortran-Compilers für die Cora 2, der Ende 1974 fertig war, wie folgt:

„Ich arbeitete bis 1971 als Systemprogrammierer im Fides-Rechenzentrum, Zürich (Tochter der damaligen Schweizerischen Kreditanstalt), bei Dr. Max Engeli (heute Prof. em. ETH). Das 1968 in Betrieb genommene Rechenzentrum war mit einer CDC 6600 ausgestattet. Ich unterstützte auch Kunden, wenn sie Probleme mit ihren Programmen hatten. Einer der Auftraggeber war Dominic Gsell, Mitarbeiter von Contraves. Er liess Fortranprogramme auf der CDC 6600 der Fides laufen. Eines Tages erzählte er mir vom Computer, der von Contraves entwickelt und hergestellt wurde. Cora 2 war sein Name. Die Maschine wurde hauptsächlich in der Flugabwehr eingesetzt, sie musste also militärtauglich sein. Angeblich konnte sie ohne Schaden einen Sturz von einem Lastwagen überstehen. Die Schweizer Armee benötigte einen Computer, der in der Schweiz gefertigt werden musste. Denn nur so war man vom Ausland unabhängig. – Woher kamen aber die Transistoren? Die Cora wurde zusammen mit dem hoch präzisen Zeichentisch auch in der Kartografie, für das Grundbuch (Kataster, Finanzministerium, Paris) und für das Zeichnen von Schnittmustern verwendet.

Dominic Gsell erklärte mir, dass Contraves dringend einen Fortran-IV-Compiler haben sollte, um die Arbeit der Ingenieure zu erleichtern. Er wollte mich für die Entwicklung des Compilers zu Contraves holen. Ich zog offenbar Nutzen aus dem Ruf meines Chefs Dr. Engeli, eines Experten für Programmiersprachen und Compiler.

Compilerbau in Europa: eine einmalige Gelegenheit

In Europa einen Compiler zu schreiben war eine einmalige Chance für einen Europäer. Die Aufgabe reizte mich sehr, aber ich hatte bisher für Max Engeli Fehler in seinem Symbal-Interpreter verbessert und Teilarbeiten bei der Entwicklung und Erweiterung ausgeführt. Wie man einen Compiler baut, wusste ich nicht. Ich zögerte zuerst, aber dann gelang es Dominic Gsell, mich abzuwerben.

Die Cora 2 hatte die Dimension einer Waschmaschine

Als ich 1971 meine Stelle antrat, hatte die Cora einen Magnetkernspeicher (an seine Kapazität erinnere ich mich nicht mehr, es war jedenfalls nicht viel), einen Lochstreifenleser, einen Lochstreifenlocher und eine Kugelkopfschreibmaschine als Ein- bzw. Ausgabegeräte. Das Bedienfeld der Frontabdeckung wies zwei Reihen binärer Schalter auf, eine für die Eingabe einer absoluten Speicheradresse und eine für die Eingabe der Daten. Nach dem Einschalten der Maschine musste man über diese Schalter ein Urladeprogramm eingeben, um die Cora zu starten.

Verfügbar war auch ein "Assembler", ein Programm, das mnemonischen (merkfähigen, einprägsamen) Text mit absoluten Adressen in binäre Maschinenbefehle umwandelte. Alles mit absoluten Speicheradressen. Und das Betriebssystem? Es gab keines. Als erstes lud man den Assembler mittels Lochstreifen in den Speicher. Dann setzte man sich an die Konsolenschreibmaschine und gab

sein Programm in Assemblersprache ein. Gleichzeitig wurde das Maschinenprogramm auf Lochstreifen ausgestanzt. Dann wurde der Assembler-Lochstreifen wieder aufgerollt, die Cora neu hochgefahren und der Programmlochstreifen in den Leser gelegt. Die Cora führte anschliessend das Programm ab Lochstreifen aus und schrieb die Ergebnisse auf der Schreibmaschine.

Um einen Compiler zu schreiben, braucht man einen richtigen Assembler. Peter Schicker, damals Assistent an der ETH, schrieb in Rekordzeit einen Assembler für die Cora, der einer CDC 6600 würdig gewesen wäre.

Um Fortranprogramme sinnvoll nutzen zu können, sind ein Betriebssystem und zeitgemässe Einrichtungen notwendig: Plattenspeicher, Lochkartenleser, Drucker. Mein blinder Kollege Arnold Schneider (Mathematiker) schrieb ein minimales Disk Operating System (Platte-Betriebssystem) mit Plattenverwaltung und Treiber für die Peripheriegeräte. Unterdessen begann ich, den Compiler zu entwerfen. Zum Compilerbau gab es schon damals eine Fülle von Schriften. Aber wie entwickelt man einen Compiler, der Fortran garantiert richtig in Maschinencode übersetzt? Optimierung war zur Zeit der noch ziemlich langsamen Computer ein wichtiges Thema. Die Anforderungen: (1) Ein Compiler sollte in möglichst kurzer Zeit übersetzen. (2) Er sollte auch ein möglichst zeit- und speicheroptimales Maschinenprogramm erzeugen. Eine Optimierung geht auf Kosten der anderen. Viel war in der Fachliteratur zu diesem Thema zu lesen. Konkrete Vorgehensweisen fand ich hingegen keine. In einem Gespräch mit meinem Vorgesetzten, der den Compiler in etwa drei Monaten erwartete, entdeckte ich eine dritte Optimierung: die Entwicklungszeit. Sie war wichtiger und dringender als die beiden anderen Optimierungsarten.

So beschloss ich, rasch ans Werk zu gehen. Um die Optimierung von Kompilationszeit und Maschinencode wollte ich mich erst dann kümmern, wenn ich eine erste korrekt laufende Version haben würde. Das erwies sich als richtige Strategie. Der Compiler übersetzte schneller, als der Drucker die Auflistung erstellen konnte. Für die Beurteilung der Ausführungsgeschwindigkeit für das Maschinenprogramm fehlten Vergleichsmöglichkeiten. Die Benutzer waren damit jedenfalls zufrieden.“

François Louis Nicolet

31.1.2013

Erläuterungen

Assembler	Assemblierer. Programm, das die in einer Assemblersprache geschriebenen Programme in (direkt) ausführbare Maschinensprache umwandelt. Assemblerprogramme zeichnen sich durch kurze Rechenzeit und geringen Speicherbedarf aus, sie sind aber auf bestimmte Rechnerarten zugeschnitten und nur schwer übertragbar. Umwandlung von symbolischen Befehlen in Maschinenbefehle und von symbolischen Adressen in Maschinenadressen
Assemblersprache	niedrige, maschinennahe Programmiersprache
Code	Programmcode, Programmtext. In einer Hochsprache geschriebenes nicht ablauffähiges Quellprogramm (Quellcode) oder in einer Maschinensprache vorliegendes ablauffähiges Maschinenprogramm (Maschinencode)
Compiler	Kompilierer. Programm, das den <i>gesamten</i> Quellcode der in einer Hochsprache geschriebenen Programme in Maschinensprache (Objektcode) übersetzt. Das Programm wird erst nach der vollständigen Übersetzung ausgeführt. Einmalige und dauerhafte Umwandlung des Quellcodes. Im Vergleich zum Interpreter wesentlich schnellere Programmausführung. Programmbetrieb ohne Compiler
Compilersprache	Programmiersprache, bei der <i>alle</i> Anweisungen, d.h. der <i>gesamte</i> Quellcode, vor der Ausführung (mithilfe eines Compilers) in Maschinencode übersetzt wird. Beispiele: Fortran, Pascal, C.

Interpreter	Interpretierer. Programm, das jede Anweisung der in einer Hochsprache geschriebenen Programme (Quellcode) zur Laufzeit <i>einzel</i> n in Maschinensprache übersetzt und unmittelbar ausführt. Die übersetzten Maschinenbefehle werden nicht aufbewahrt. Zeilenweise Abarbeitung des Quellcodes. Im Vergleich zum Compiler einfachere Veränderung und Erprobung von Programmen. Programmbetrieb nur mit Interpreter.
Interpretersprache	Programmiersprache, bei der der Quellcode (mithilfe eines Interpreters) Befehl für Befehl übersetzt und ausgeführt wird. Beispiele: Basic, Logo, viele Makrosprachen
Maschinencode	Maschinensprache, Folge von Einsen und Nullen (Binärdarstellung), vom Computer direkt verarbeitbare Programmiersprache
Maschinensprache	Maschinencode, Folge von Einsen und Nullen (Binärdarstellung), vom Computer direkt verarbeitbare Programmiersprache, niedrigste Programmiersprache
Objektcode	Programmcode (in Maschinensprache), der von einem Assembler oder einem Compiler durch die Übersetzung des Quellcodes erzeugt wird, nur bedingt (in Verbindung mit Unterprogrammbibliotheken) lauffähig
Quellcode	Programmcode (in Hochsprache), Quelltext, Quellprogramm. Programmanweisungen, die in Assembler oder einer höheren Programmiersprache geschrieben sind. Menschen können die Befehle lesen, die Maschine kann sie erst nach Übersetzung in die Maschinensprache verarbeiten

Quellen: mehrere Informatiklexika

*) pensionierter Dozent der ETH Zürich für Fachdidaktik Informatik

Buchhinweis

Herbert Bruderer: Konrad Zuse und die Schweiz. Wer hat den Computer erfunden? Oldenbourg-Verlag, München 2012, XXVI, 224 Seiten

