

Komorowski, Witold

Pierwsze polskie komputery Odra

Kwartalnik Historii Nauki i Techniki 47/2, 97-110

2002

Artykuł umieszczony jest w kolekcji cyfrowej Bazhum, gromadzącej zawartość polskich czasopism humanistycznych i społecznych tworzonej przez Muzeum Historii Polski w ramach prac podejmowanych na rzecz zapewnienia otwartego, powszechnego i trwałego dostępu do polskiego dorobku naukowego i kulturalnego.

Artykuł został zdigitalizowany i opracowany do udostępnienia w internecie ze środków specjalnych MNiSW dzięki Wydziałowi Historycznemu Uniwersytetu Warszawskiego.

Tekst jest udostępniony do wykorzystania w ramach dozwolonego użytku.



Witold Komorowski

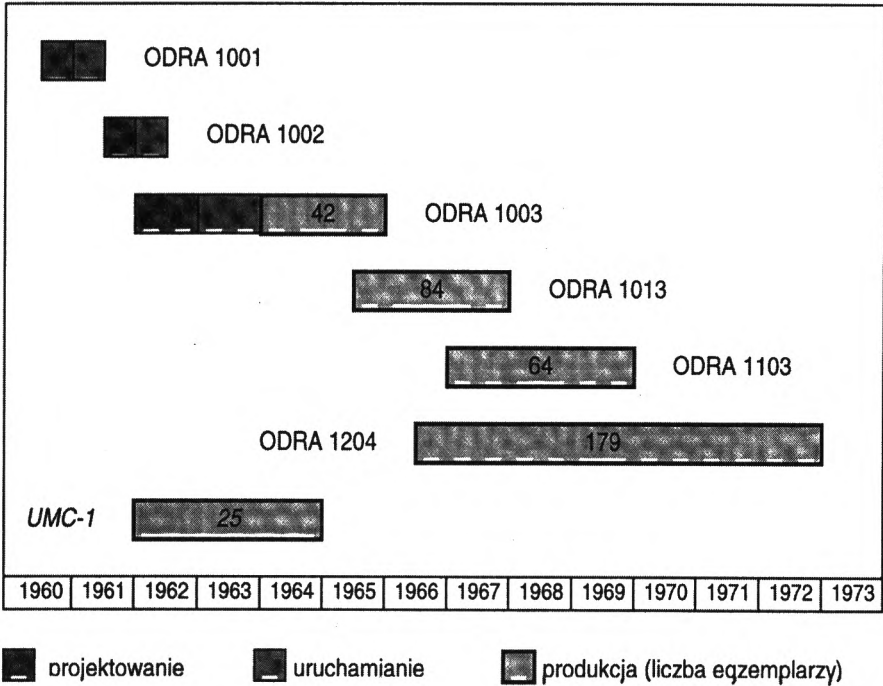
Wydziałowy Zakład Informatyki PWr.
Wrocław

PIERWSZE POLSKIE KOMPUTERY *ODRA*

Od kilku lat trwają mniej lub bardziej formalne obchody różnych okrągłych rocznic związanych z techniką informatyczną. I tak, w roku 1996 obchodzono „50-lecie komputerów”, gdyż mijało pół wieku od oficjalnej prezentacji legendarnego *ENIAC-a* (*Univeristy of Pennsylvania*, 14 lutego 1946) oraz 25 lat od wyprodukowania pierwszego mikroprocesora Intel 4004 (1971). Trzy lata później przypadała rocznica uruchomienia pierwszego „prawdziwego” komputera (z programem umieszczonym w pamięci wewnętrznej) - brytyjskiego *ED-SAC-a* (*University of Cambridge*, 6 maja 1949). W roku 2001 minęło 50 lat od uruchomienia maszyny *UNIVAC* – pierwszego komputera produkowanego seryjnie (Filadelfia, 14 czerwca 1951). W kontekście tych rocznic warto przypomnieć polskie początki informatyki i przemysłu „maszyn matematycznych”. Interesująca jest nie tylko historia zespołów ludzkich i działań organizacyjnych [2, 11], ale też koncepcje techniczne związane z ówczesnym stanem wiedzy i możliwościami technologicznymi.

W latach 60. we Wrocławskich Zakładach Elektronicznych „Elwro” zaprojektowano i wytwarzano maszyny cyfrowe nazywane *ODRA*. W stosunku do tych maszyn stosowano wówczas, a nawet i później, określenie „rodzina komputerów”, co można rozumieć jedynie w sensie przenośnym ponieważ miały wspólną genealogię, natomiast nie były kompatybilne – różniły się nie tylko realizacją techniczną i organizacją logiczną, ale przede wszystkim architekturą. Rodzinę maszyn tworzyły dopiero późniejsze, produkowane od 1970 r., modele serii *Odra 1300* zgodne programowo (kompatybilne) również z angielskimi komputerami *ICL 1900*.

Nota bene moda na tworzenie nazw komputerów od rzek bliskich sercu projektantów nie była tylko wrocławską specjalnością. Jedyny znany (z lakonicznych przekazów) komputer zbudowany z elementów 3-stanowych, *Sietuń*, wziął nazwę od małego dopływu rzeki Moskwy przepływającego nieopodal Uniwersytetu Łomonosowa. Podobnie wyglądała sprawa 40 lat później i w zupełnie innej części świata – od lat oczekiwany 64-bitowy procesor *Intel-a*, ogłoszony ostatecznie pod firmową nazwą *Itanium*, początkowo był znany jako *Merced* – od Rio Nuestra Senora de la Merced płynącej wśród gór Parku Narodowego Yosemite.



Rys.1. Produkcja komputerów w WZE „Elwro” w latach 60. XX wieku.

Na diagramie (rys.1) przedstawiono w układzie chronologicznym produkcję pierwszych komputerów *ODRA* [1,2,14], będących w latach 60. oryginalnymi konstrukcjami „Elwro”. Umieszczono tam również maszynę *UMC-1*, która została wprowadzie zaprojektowana na Politechnice Warszawskiej, ale dzięki przeniesieniu do Wrocławia stała się pierwszą polską maszyną produkowaną przemysłowo. Podaną wielkość produkcji należy odnieść do ówczesnego stanu posiadania – w roku 1965 użytkowano w Polsce 60 komputerów, a w 1970 ta liczba wzrosła do... 211 [13].

W komputerach *ODRA* powtórzono etapy pionierskiego okresu komputerów z lat 40. i 50.; dotyczy to zarówno technik realizacji, organizacji logicznej, jak i architektury (tab.1).

Tab. 1. Charakterystyka maszyn ODRA

	Organizacja	Układy	Pamięć	Długość słowa	Uwagi
Odra 1001	Szeregowa	Tranzystorowo-transformatorowe oraz lampowe	512 słów, bębnowa	17 bitów (17+1)	Model [1,5]
Odra 1002	Szeregowa	Tranzystorowo-transformatorowe	4096 słów, bębnowa	35 bitów (35+1)	Działający model; programowanie w języku wewnętrznym [16]
Odra 1003	Szeregowa	Tranzystorowo-transformatorowe	8192 słów, bębnowa	39 bitów (39+1)	Pierwsza produkowana przemysłowo (42 egz.); autokod MOST 1
Odra 1013	Szeregowa	Tranzystorowo-transformatorowe	7 936 słów, bębnowa + 256 słów, ferrytowa	39 bitów (39+1)	Ulepszona wersja Odry 1003 (zastosowanie pamięci ferrytowej) [4]
Odra 1103	Szeregowa	Tranzystorowo-transformatorowe	1024 słowa, ferrytowa	16 bitów	Kalkulator dziesiętny (w kodzie BCD) do współpracy z maszynami analitycznymi [10]
Odra 1204	Równoległa	Tranzystorowe	65 536 słów, ferrytowa	24 bity	Sterowana mikroprogramem z pamięci ROM; język ALGOL 1204; system MASON [6,14]

PAMIĘĆ

Wydaje się, że podstawowym czynnikiem wpływającym na koncepcję tych maszyn była realizacja pamięci operacyjnej jako pamięci bębnowej. W takiej pamięci informacja jest przechowywana na powierzchni magnetycznej pokrywającej obracający się bęben. Do odczytu i zapisu służą nieruchome głowice, które odczytują bądź zapisują sekwencyjnie kolejne bity słowa. Każda głowica wyznacza swoją ścieżkę zawierającą zwykle kilkadziesiąt słów, do których dostęp jest cykliczny. Pamięci bębnowe wprowadzono w latach 1947–1948 i były stosowane, niekiedy równoległe z pamięciami na lampach elektrostatycznych (F.C. Williams w maszynie *Manchester Mark I*), aż do czasu upowszechnienia pamięci na rdzeniach ferrytowych, których praktycznym debiutem było ich zastosowanie przez J.W.Forrestera w 1952 w komputerze *Whirlwind* zbudowanym

w MIT. Stosowanie pamięci bębnowej prowadziło do koncepcji maszyn niesekwencyjnych – wykonujących rozkazy w innej kolejności niż wynikająca z kolejności ich adresów. W maszynie niesekwencyjnej rozkaz zawiera oprócz adresów argumentu również adres następnego rozkazu; np. pierwsza von Neumannowska maszyna *EDVAC* była (3+1)-adresowa, maszyny *ODRA* były „jeden plus jeden adresowe”. Maszyny niesekwencyjne umożliwiały optymalne rozmieszczenie rozkazów w pamięci: jeżeli jest znany czas wykonania rozkazu, to rozkaz następny należy umieścić pod adresem, który będzie dostępny do odczytu w momencie zakończenia bieżącego rozkazu. Wszystkie maszyny *ODRA* aż do modelu 1013 (również *UMC-1*) miały pamięć bębnową i wszystkie, z wyjątkiem modelu 1001, były niesekwencyjne. Ponieważ maszyny te były programowane w języku wewnętrznym (kody rozkazów zapisywane ósemkowo), w ich dokumentacji można znaleźć odpowiednie tabele podające długość cyklu rozkazowego, co ułatwiało optymalne programowanie.

W pierwszym zaprojektowanym komputerze *ODRA 1001* przyjęto prostszą organizację – sekwencyjną (słowo jest jedno-adresowe). W modelu tym, jako jedynym w omawianej serii, oprócz pamięci bębnowej zastosowano dwa układy pamięci na liniach opóźniających do realizacji czterech tzw. „szybkich” rejestrów. Pamięci tego typu były stosowane w pierwszych historycznych komputerach, np. w amerykańskich *ENIAC* i *EDVAC*, czy angielskim *EDSAC* – pierwszej w ogóle maszynie z pamiętanym programem. W maszynach tych jako linię opóźniającą stosowano rury wypełnione rtęcią (*mercury acoustic delay line*), w których za pomocą przetworników piezoelektrycznych wzbudzano drgania akustyczne przenoszone na drugi koniec, tam odczytywane i ponownie wprowadzane do linii. Maszyna *EDSAC* (1949) wzięła nawet swoją nazwę od zastosowanej pamięci (*Electronic Delayed Storage Automatic Calculator*). W maszynie *ODRA 1001* zastosowano bardziej poręczną pamięć magnetostrykcyjną; drgania mechaniczne wzbudzano tu w cienkiej rurce niklowej za pomocą przetworników magnetostrykcyjnych. Pamięci takie, mimo że miały dostęp cykliczny, rzeczywiście były szybsze niż pamięć bębnowa ze względu na mniejszą liczbę słów zapamiętanych w linii i większą szybkość cyrkulacji bitów (szybkość fali akustycznej w niklu wynosi około 5 m/ms). Pamięć ferrytową wprowadzono po raz pierwszy w modelu *ODRA 1013* kompatybilnym z wcześniejszym *1003*. Różnica między nimi polegała na zastąpieniu dwóch spośród 64 ścieżek pamięci bębnowej – łącznie 256 słów – pamięcią ferrytową o dostępie swobodnym (*random access*). Ta, pozornie niewielka, zmiana pozwoliła zwiększyć szybkość obliczeń pięciokrotnie (średni czas dostępu w pamięci bębnowej wynosił około 11 ms, czas cyklu pamięci ferrytovej 8 μ s). Dopiero *ODRA 1204* miała całą pamięć typu RAM zrealizowaną na rdzeniach ferrytowych. Zastosowano rdzenie toroidalne o średnicy zewnętrznej 1,3 mm przełączane przez koincydencję prądów przepływających przez dwa przewody wybierające; trzeci przewód przechodzący

przez rdzeń był przewodem odczytu, zaś czwarty – przewodem zakazu stosowanym przy zapisie. Rdzenie zestawiono w ramkach 64×64 zawierających po jednym bicie każdego z 4096 słów. Cztery bloki po 25 ramek dawały łączną pojemność pamięci 16 K słów 25-bitowych. Czas dostępu wynosił 2 μ s, a czas cyklu 6 μ s.

UKŁADY LOGICZNE

We wszystkich maszynach ODRA od modelu 1001 aż do 1103 stosowano podobną technikę realizacji układów logicznych odpowiednią do szeregowego przetwarzania słów binarnych. Funkcje logiczne (AND, OR, NAND, NOT) realizowano statycznie w technice diodowo-tranzystorowej, natomiast konieczne opóźnienie i synchronizację zapewniały tzw. układy opóźniające zbudowane jako wzmacniacz tranzystorowy z obciążeniem transformatorowym. Układ ten, poza dwoma napięciami stałymi, był dodatkowo zasilany impulsowo dwoma specjalnie ukształtowanymi ciągami zegarowymi [3,7]. Była to tranzystorowa wersja tzw. układu Havensa stosowanego w wersji lampowej w maszynie IBM 701 (1953). Przyjęcie tak skomplikowanego układu elektronicznego było uzasadnione potrzebą zapewnienia odpowiedniej niezawodności działania całej sieci logicznej, gdyż same elementy (diody i tranzystory germanowe), mimo wstępnej selekcji i starzenia, miały duże rozrzuty parametrów. Warto zauważyć, że przyjęty do produkcji w "Elwro" w latach 1962–1964 komputer UMC-1 był zrealizowany również w technice dynamicznej, ale z zastosowaniem lamp próżniowych (układ opóźniający podobny do tzw. układu Owen'a [12,5]). Dzięki zastosowaniu układu opóźniającego pamiętanie informacji odbywało się dynamicznie – w pętli sprzężenia zwrotnego, tworząc tzw. przerzutnik dynamiczny (jednobitowy) lub w cyklicznym rejestrze przesuwym (wielobitowym). Jedynek logiczną reprezentował impuls ujemny (-9 V), zero logiczne – brak impulsu (około 0 V).

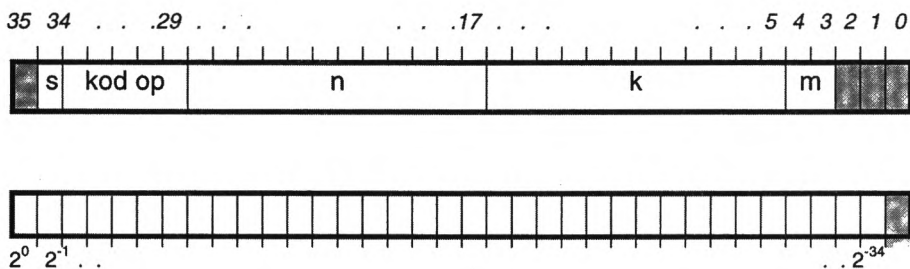
ODRA 1002

Reprezentatywnym przykładem organizacji i architektury pierwszych maszyn typu ODRA był model 1002. Był to komputer pozbawiony uproszczeń modelu 1001 [17], choć zbudowany według podobnej koncepcji; przede wszystkim była to pierwsza działająca maszyna, dla której istniało podstawowe oprogramowanie i na której uruchamiano programy użytkowe. Doświadczenie uzyskane przy projektowaniu i uruchamianiu tej maszyny, a także przygotowaniu oprogramowania dla niej stało się podstawą niewątpliwego sukcesu jej bezpośrednich następców – ODRY 1003 i 1013. Wszystkie one były (1+1)-adresowe, miały podobną podstawową listę rozkazów i były zbudowane z podobnych układów

logicznych. Szeregowy dostęp do pamięci (bit po bicie) narzucił szeregowy sposób przetwarzania informacji. Cykl zegara wyznaczał czas przesłania (i przetworzenia) jednego bitu; czas ten określano jako „chwilę maszynową”, wielokrotność „chwili” wystarczająca na przesłanie całego słowa była nazwana „krokiem”. Dla zachowania synchronizacji z pamięcią impulsy zegarowe były generowane ze specjalnej ścieżki na bębnie, dzięki czemu niewielkie wahania obrotów nie powodowały błędów. Komunikacja procesora z pamięcią wymagała specyficznego traktowania adresów – każdy adres dzielił się na część „przestrzenną” odpowiedzialną za wybór właściwej ścieżki i część „czasową” określającą moment odczytu lub zapisu słowa z wybranej ścieżki na bębnie. Układ sterujący był zrealizowany jako automat skończony synchronizowany zegarem. W cyklu rozkazowym zarówno w fazie pobierania rozkazu jak i w fazie wykonania występowały okresy „oczekiwania na adres”, których wielkość zależała od względnego przesunięcia między adresem rozkazu i adresem jego argumentu oraz od położenia kolejnych rozkazów programu. Mimo iż zastosowano bardzo prosty arytmometr (jeden(!) układ sumatora 1-bitowego na wejściu przesuwnej rejestru akumulatora) *ODRA 1002* miała zrealizowane układowo operacje mnożenia i dzielenia stałoprzecinkowego.

SŁOWO MASZYNOWE

Słowo w pamięci (rys.2) było 36-bitowe, przy czym bit $\langle 0 \rangle$ (w czasie szeregowego odczytu pojawiający się jako pierwszy) był traktowany jako „bit techniczny” i nie brał udziału w operacjach. Słowo rozkazowe było 32-bitowe (bity $\langle 34:3 \rangle$ słowa pamięci). Słowo liczbowe było 35-bitowe i oznaczało liczbę dwójkową w kodzie uzupełnieniowym (U2) z zakresu $-1 \leq x < +1$ (przecinek dwójkowy po bicie najbardziej znaczącym).



Rys. 2. Umieszczenie słowa rozkazowego i słowa liczbowego w 36-bitowym słowie maszyny *ODRA 1002*.

Znaczenie poszczególnych pól słowa rozkazowego jest następujące:

- s – bit stopu; gdy $s=1$ i na pulpicie jest wciśnięty klawisz „stop R”, to po wykonaniu rozkazu następuje zatrzymanie; ponowne uruchomienie jest możliwe tylko z pulpitu;
- kod operacyjny – zdefiniowano działanie dla każdego z 32 możliwych kombinacji kodowych, chociaż w opisie zalecano stosowanie tylko 27 kodów (zob. lista rozkazów);
- n – adres argumentu w rozkazach komunikujących się z pamięcią,
 - parametr w rozkazach przesunięć lub
 - adres skoku efektywnego w rozkazach warunkowych SkW i SkN;
- k – adres następnego rozkazu; adres ten jest przekazywany do rejestru adresowego (aR) na początku cyklu rozkazowego;
- m – indeks modyfikacji; określa sposób modyfikacji następnego rozkazu (brak modyfikacji, dodanie do części n zawartości rejestru F, dodanie do całego rozkazu zawartości rejestru M, dodanie do całego rozkazu sumy logicznej F or M).

Pamięć bębnowa daje dostęp szeregowy (kolejne bity słowa poczynając od bitu $\langle 0 \rangle$) i cykliczny (na jednej ścieżce rozmieszczone były kolejno 64 słowa). Rytm pracy procesora (zegar) był synchronizowany obrotami bębna pamięci; 2304 impulsy ze ścieżki zegarowej dawały przy szybkości 2800 obr/min częstotliwość 107,52 kHz. Czas przesłania jednego słowa wynosił więc około 336 mikrosekund, co daje szybkość około 300 słów/s.

LISTA ROZKAZÓW

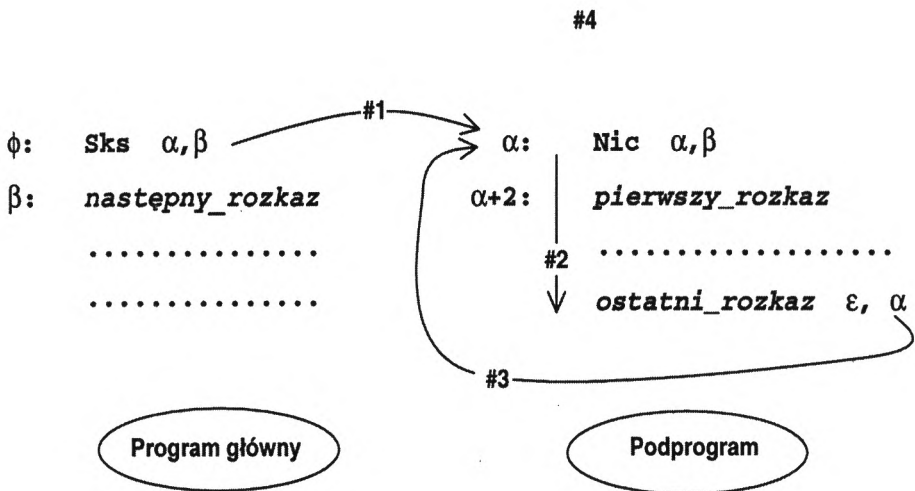
Lista rozkazów maszyny ODRA 1002 (tab.2.) jest typowa dla maszyn niesekwencyjnych jednoargumentowych i oprócz rozkazów spotykanych we współczesnych maszynach ma też działania, które obecnie wydają się nieco egzotyczne. Głównym rejestrem arytmometru jest tu akumulator A, który zawsze zawiera domyślny argument działań 2-argumentowych. Drugi rejestr arytmometru M spełnia pomocniczą rolę przy mnożeniu i dzieleniu. Rozkazy przesłań typu *load* i *store* zapewniają komunikację tych rejestrów z pamięcią (doA, doM, zA, zM). Do akumulatora można ponadto wpisywać wartość ujemną wskazanego argumentu (odA) i zero (ZrA). Ostatni rozkaz z grupy przesłań ładuje parametr n rozkazu do krótkiego rejestru modyfikacji F. Rozkazy przesunięć (Ap, Cp, Al) dotyczą zawsze 35-bitowej zawartości akumulatora, przy czym liczba pozycji jest podawana jako parametr. Rozkazy arytmetyczne działające na argumentach w kodzie uzupełnieniowym (U2) realizują cztery podstawowe działania (Do, Od, Mn, Dz) oraz znajdowanie wartości bezwzględnej (Bw) i liczby przeciwnej (uza). Mnożenie i dzielenie jest wykonywane układowo, co zajmuje 35 kroków i w działaniach tych bierze udział rejestr M (mniej znacząca część

Tab. 2. Odra 1002 – Lista rozkazów

Kod operacyjny (ósemkowo)	Symbol	Nazwa	Działanie	Liczba kroków w fazie wykonania
00*		<i>Nic nie rób</i>	–	1
01	Sks	Skok ze śladem	$B[n] \leftarrow 37_8^n k^m$; next aR $\leftarrow n+2$	1
02*		<i>Nic nie rób</i>	–	1
03*		<i>Nic nie rób</i>	–	1
04	Mn	Mnożenie	$A^M \leftarrow A * B[n]$	35
05	Dz	Dzielenie	$A \leftarrow A / B[n]$; $M \leftarrow A \bmod B[n]$	max 37
06	doM	Przesłanie do M	$M \leftarrow B[n]$	1
07	zM	Przesłanie z M do pamięci	$B[n] \leftarrow M$	1
10	Kon	Koniunkcja	$A \leftarrow A \text{ and } B[n]$	1
11*		<i>Zerowanie A</i>	$A \leftarrow 0$	1
12	Rs	Różnica symetryczna	$A \leftarrow A \text{ xor } B[n]$	1
13	zA	Przesłanie z A do pamięci	$B[n] \leftarrow A$	1
14	doA	Przesłanie do A	$A \leftarrow B[n]$	1
15	oda	Minus przesłanie do A	$A \leftarrow \text{neg } B[n]$	1
16	Do	Dodawanie	$A \leftarrow A + B[n]$	1
17	Od	Odejmovanie	$A \leftarrow A - B[n]$	1
20	We	Wejście	$A<35:31> \leftarrow \text{We}$	17
21	Wy	Wyjście	$\text{Wy} \leftarrow A<35:31>$	129
22	SkW	Skok przy warunku W	$(W=1) \Rightarrow aR \leftarrow n$	1
23	SkN	Skok przy warunku N	$(N=1) \Rightarrow aR \leftarrow n$	1
24	Bw	Bezwzględna wartość	$A \leftarrow \text{abs } A$	1
25	uzA	Uzupełnienie	$A \leftarrow \text{neg } A$	1
26	Stop	Stop z pobraniem	$aR \leftarrow k$; next stop	1
27*		<i>Nic nie rób</i>	–	1
30	Ap	Arytmetyczne przesunięcie w prawo	$A \leftarrow n \text{ ashr } A$	n
31	Cp	Cykliczne przesunięcie w prawo	$A \leftarrow n \text{ ror } A$	n
32	Al	Arytmetyczne przesunięcie w lewo	$A \leftarrow n \text{ shl } A$	n
33	Okr	Zaokrąglenie	$A \leftarrow A + 2^{-35}$	1
34	ZrA	Zerowanie A	$A \leftarrow 0$	1
35*		<i>Zerowanie A</i>	$A \leftarrow 0$	1
36	doF	Przesłanie do F	$F \leftarrow n$	1
37	Nic	<i>Nic nie rób</i>	–	1

iloczynu lub reszta z dzielenia). Ze względu na przyjętą „skalę zerową” (przecinek jest umieszczony po najbardziej znaczącym bicie liczby) sensowna jest operacja zaokrąglania (Okr) wyniku ułamkowego. Są tylko dwa rozkazy operacji logicznych – koniunkcja (Kon) i alternatywa wyłączająca (Rs) – działających na odpowiadających sobie bitach akumulatora i wybranego słowa w pamięci. Ze względu na niesekwencyjne wykonywanie rozkazów specyficzna jest realizacja skoków. Ponieważ każdy rozkaz zawiera adres następnego rozkazu, nie ma potrzeby stosowania skoków bezwzględnych. Dwa skoki względne (SkN i SkW) badają stan znaczników N i W ustawianych po każdej operacji (za wyjątkiem przesłań i samych skoków).

Oryginalną konstrukcję ma wywołanie podprogramu, które odbywa się rozkazem „skok ze śladem” (Sks). Rozkaz ten wpisuje pod wskazanym adresem n kod rozkazu „nic nie rób” z adresem następnego rozkazu, takim, jaki był w Sks a następnie przechodzi do rozkazu z komórki $n+2$, gdzie powinien znajdować się pierwszy rozkaz podprogramu. Wybór $n+2$ jako adresu kolejnego rozkazu był podyktowany chęcią optymalizacji czasu wykonania: wykonanie rozkazu Sks, czyli zapis do komórki n zajmuje 1 krok, po którym gotowa do odczytu (w tym przypadku celem pobrania rozkazu) jest komórka $n+2$. Powrót do programu głównego następuje przez skok do adresu n symulując w ten sposób adresację pośrednią; podprogramy muszą więc być tak konstruowane, by ich pierwszy adres zawierał „przechowalnię” adresu powrotnego. Stosując uwspółcześniony zapis symboliczny, można tę sytuację zapisać jak w poniższym przykładzie (uwaga: ϕ i β nie są kolejnymi adresami!):



Rozkazy wejścia-wyjścia (We i Wy) są specjalizowane – dotyczą dwóch konkretnych urządzeń zewnętrznych przewidzianych do współpracy z *ODRA 1002*: czytnika i perforatora taśmy papierowej. Dodatkowym urządzeniem był dalekopis spełniający w istocie rolę dekodera z kodu 5-bitowego na postać alfanumeryczną i odwrotnie. Czytnik dostarczał 5-bitowy kod znaku na pozycje <35:31> akumulatora z maksymalną szybkością 300 znaków/s, perforator wyprowadzał na taśmę kod z tych samych pozycji akumulatora z szybkością 30 znaków/s. Z tych szybkości wynikał czas przeznaczony na każdy z rozkazów wejścia-wyjścia, odpowiednio 17 i 129 kroków.

Pełną listę rozkazów przedstawia tabelka, w której przyjęto oznaczenia:

- B[0:4095]<35:0> – pamięć operacyjna o pojemności 4 Ksłów 36-bitowych;
- aR <11:0> – adres następnego rozkazu do wykonania;
- A<35:0> – akumulator (w działaniach bierze udział 35 bitów <35:1>);
- M<35:0> – rejestr uniwersalny mnożnika i modyfikacji (w działaniach bierze udział 35 bitów <35:1>);
- F<25:14> – rejestr modyfikacji części n słowa rozkazowego;
- N, W – jednobitowe znaczniki stanu (po operacjach arytmetycznych oznaczają nadmiar i wynik ujemny).

Umieszczone w tabelce symbole rozkazów nie są mnemonikami w dzisiejszym rozumieniu, ponieważ nie można ich było stosować w programach; *ODRA 1002* była programowana w języku wewnętrznym (nie symbolicznym) a jedynym ułatwieniem było stosowanie zapisu ósemkowego zamiast dwójkowego przy kodowaniu rozkazów i adresów. Podane czasy wykonania dotyczą tylko fazy wykonania; pełny cykl rozkazowy zawierał czas oczekiwania na rozkaz (max 63 kroki), czas pobrania rozkazu (1 krok), czas oczekiwania na argument (max 63 kroki) i czas wykonania. Choć *ODRA 1002* nie miała systemu przerwań w obecnym tego słowa znaczeniu, możliwe było zatrzymanie wykonywania programu w szczególnych sytuacjach i ewentualne ponowne uruchomienie przez operatora. Ustawianie punktów zatrzymania („*breakpointów*”), bardzo przydatne przy uruchamianiu, było możliwe dzięki bitowi s w słowie rozkazowym – jeżeli na pulpicie był wciśnięty przycisk „StopR” wykonywanie zostawało zatrzymane po wykonaniu rozkazu z s=1. Podobną pułapkę przewidziano dla warunku N (nadmiar w rozkazach arytmetycznych) – praca była zatrzymywana jeżeli wystąpiło N=1 i na pulpicie był wciśnięty przycisk „StopN”. Bezwzględne zatrzymanie pracy możliwe było rozkazem „stop z pobraniem”, który przygotowywał w rejestrze aR adres następnego rozkazu umożliwiając operatorowi kontynuację przerwanej pracy.

Tab. 3. Możliwości operacyjne maszyn ODRA

Maszyna	Odra 1001	Odra 1002	Odra 1003	Odra 1013	Odra 1103	Odra 1204
Liczba rozkazów	28	26 (32)	259	263	40	108
Szybkość (dodawania/s)	200	800	500	1 000	5 000	60 000
Działania zmp.	Nie	Programowo	Układowo	Układowo	Tylko dziesiętne	Układowo

ŚLADY HISTORII

Przypomnienie pierwszych polskich maszyn wydaje się celowe nie tylko dla uczczenia jubileuszowych rocznic – minęło akurat 40 lat od uruchomienia *ODRY 1001* – ale jest też pouczające. Analiza tych konstrukcji pokazuje prze-
 możny wpływ sposobu realizacji technicznej, która jest najszybciej zmieniającą się cechą komputerów, na cechę najbardziej konserwatywną – architekturę logiczną, z którą bezpośrednio styka się programista [8]. W wielu miejscach na świecie rekonstruowane są pionierskie rozwiązania techniczne (np. maszyna Atanasoffa – sędownie (!) uznana w USA za pierwszy automatyczny komputer elektroniczny) lub są tworzone ich wirtualne modele (zob. liczne odnośniki np. w [15]). Wydaje się, że trzeba by podobną działalnością „muzealniczą” objąć rodzime produkty, gdyż już obecnie trudno nie tylko o fizyczne eksponaty, ale nawet o ich dokumentację techniczną.

Bibliografia

1. *100 maszyna matematyczna z WZE Elwro*. „Biuletyn Zakładowy Elwro”. Wydanie specjalne. Wrocław, listopad 1966.
2. E. B i l s k i : *Okres maszyn cyfrowych typu ODRA*. „Informatyka” 1989 nr 8–12.
3. J. B r o m i r s k i , A. S i e l i c k i , A. Z a s a d a : *Tranzystorowa realizacja sieci logicznych w oparciu o układ opóźniający Havensa z użyciem elementów produkcji krajowej*. „Biuletyn Informacyjny Elwro”. Nr. 2. Wrocław 1962.
4. *Dokumentacja Techniczno-Ruchowa Maszyny Cyfrowej ODRA 1013. Opis ogólny*. Wrocław 1965 WZE Elwro.
5. I. H. G o u l d , F. S. E l l i s : *Digital Computer Technology*. London 1963 Chapman & Hall Ltd.
6. Th. K a m b u r e l i s : *Opis funkcjonalny maszyny cyfrowej Odra 1204*. Wrocław 1967 WZE Elwro.

7. W. K o m o r o w s k i : *Synteza sieci logicznych w oparciu o przerzutnik dynamiczny*. „Zeszyty Naukowe Politechniki Wrocławskiej Automatyka” I. Nr 85. Wrocław 1964.
8. W. K o m o r o w s k i : *Instrumenta computatoria. Wybrane architektury komputerów*. Gliwice 2000 Helion.
9. J. K s i ą ż e k : *Pamięć operacyjna m.c. Odra 1204*. [w : 14].
10. J. L e s i ń s k i : *ODRA 1103. Opis ogólny*. Wrocław 1967 WZE Elwro.
11. J. M a d e y , M. M. S y s ł o : *Początki informatyki w Polsce*. „Informatyka” nr 10 2000.
12. J. M i e ś c i c k i : *Uniwersalna maszyna cyfrowa UMC-1*. W: *Laboratorium organizacji maszyn cyfrowych* pod red. K. F i a ł k o w s k i e g o . Warszawa 1968 Wydawnictwo Politechniki Warszawskiej.
13. *Prognoza rozwoju informatyki w Polsce do roku 2000*. (opracowali: A. T a r g o w s k i , S. B r a m s k i , M. R y b a k). Warszawa 1972 OBRI.
14. *Techniczna realizacja maszyny cyfrowej Odra 1204. Materiały Sympozjum*. Wrocław 1968.
15. *The Virtual Museum of Computing (VMoC)*. <http://vlmp.museophile.com>
16. R. Z u b e r , J. M a r k o w s k i , A. N i ż a n k o w s k i : *Założenia konstrukcji specjalizowanej elektronicznej maszyny cyfrowej ODRA 1001*. (maszynopis) Wrocław 1960 Elwro.
17. R. Z u b e r : *Maszyna cyfrowa „Odra 1002”. Instrukcja obsługi i programowania* (maszynopis powielany). Wrocław 1962 Elwro.

Witold Komorowski

EARLY POLISH COMPUTERS OF THE ODRA TYPE

The first Polish computers were designed and produced at the Wrocławskie Zakłady Elektroniczne „Elwro” in Wrocław in the 1960s. The machines, called ODRA, came in several models over a span of several years; in their design, the successive models of the ODRA computers emulated the pioneering stages of the development of computers in the 1940s and 1950s, relating both to implementation technology, logical organization as well as architecture (table 1). The diagram (fig. 1) presents the chronological development of the ODRA computers, as well as the UMC-1 computer designed at the Warsaw Technological University (Politechnika Wraśawska), the first Polish computer to be manufactured on an industrial scale.

The concept underlying the design of the ODRA computers seems to have been the implementation of the main memory as drum memory. In drum memory information is stored on a magnetic surface covering a revolving drum. Using drum memory led to the concept of non-sequential computers in which instructions were executed in a sequence different from that of their addresses. In a non-sequential computer an instruction contains not only operand addresses but also the address of the next instruction.

All the ODRA computers, beginning from model 1001 up to model 1103, used a similar technology of logic gates implementation, which was appropriate for the bit-by-bit processing of binary words. Logical functions (AND, OR, NAND, NOT) were implemented in diode-transistor logic, while the necessary delay and synchronization were ensured by the so-called delay elements constructed as transistor amplifiers with transformer load.

The 1002 model is very representative of the organization and architecture of the first computers of the ODRA range; it was the first operational computer that had special software developed for it and on which processing programs could be run.

In the ODRA 1002 (fig. 2), there were 36-bit words in the memory, with the instruction words being 32-bit words and data words being 35-bit words, signifying a binary number in 2's complement code in the range $-1L \times +1$ (binary point after the most significant bit).

The instruction set of the ODRA 1002 computer (table 2) is typical for non-sequential one-operand computers and apart from instructions found in modern computers it also has functions which may now seem somewhat exotic. Due to the non-sequential execution of instructions it has specific implementation of branches. Because each instruction contains the address of the next instruction, there is no need to use unconditional jumps.

The instruction symbols presented in the table are not mnemonics in the contemporary understanding of the word, because they could not be used in programs; the ODRA 1002 computer was programmed in machine code (and not in symbolic language) and the only facilitating feature was the use of octal instead of binary notation in coding instructions and addresses.

A look at the first Polish computers may be very illuminating; the analysis of those constructions shows how great an influence technological implementation, which is the most readily changeable feature of computers, may have on the most conservative feature – logical architecture, with which the programmer is directly concerned.

