

Alina MOMOT¹

¹Wydział Automatyki, Elektroniki i Informatyki, Politechnika Śląska, ul. Akademicka 16, 44-100 Gliwice

Projektowanie rozkazów dla maszyny W – konspekt ćwiczeń laboratoryjnych

Streszczenie. Artykuł opisuje propozycję przeprowadzenia ćwiczeń laboratoryjnych dotyczących podstaw projektowania rozkazów dla maszyny W. Zakłada się przy tym, że uczestnicy laboratorium dysponują programowym symulatorem maszyny W oraz przyswoili już podstawowe wiadomości na temat architektury tej maszyny, sposobu działania układu sterującego i ogólnych zasad projektowania rozkazów.

Słowa kluczowe: maszyna W, projektowanie rozkazów, konspekt zajęć.

1. Wstęp

Skutecznie przeprowadzone zajęcia lekcyjne powinny obejmować przede wszystkim prezentację zrozumienia materiału przez ucznia. Wymaga to od uczniów kierowania się celem uczenia się i pogłębiania rozumienia omawianego materiału. Prezentacja taka stanowi dowód zrozumienia treści i nabycia umiejętności zawartych w celu uczenia się [5]. Dlatego też bardzo ważne jest, aby zarówno uczeń, jak i nauczyciel znali ten cel i zmierzali w jego kierunku. Równie ważne jest określenie i przygotowanie odpowiednich kryteriów sukcesu oraz odpowiednie zwiększanie stopnia trudności realizowanych zadań w zależności od możliwości danego ucznia. Jak piszą autorki książki *Cele uczenia się. Jak pomóc uczniom zrozumieć każdą lekcję* (zob. [5], s. 76): „Bardzo ważne jest tutaj rozróżnienie celu uczenia się i oceny. Nie chodzi nam o to, że oceny nie są ważne. Uważamy jednak, że kiedy nauczyciele zachęcają uczniów do uczenia się na konkretny stopień, a nie do opanowania materiału, który spowoduje wystawienie takiego stopnia, ograniczają możliwości swoich uczniów”.

Zachęcanie uczniów do samodzielności jest kluczowym elementem procesu uczenia się. Potrzebę samodzielności w zdobywaniu wiedzy zauważano już w ubiegłym wieku. Jak pisze bowiem w wydanej w 1938 roku książce Ignacy Schreiber (zob. [7], s. 8–9): „uczeń *naprawdę* musi wprawić się w samodzielne zdobywanie wiedzy i nauczyć się samodzielnie osiągniętymi już wiadomościami operować. W przeciwnym bowiem wypadku przyjmuje gotową potrawę już pokrajaną i przygotowaną ostatecznie do spożycia, ale może ją tylko połknąć i w miarę nabywanej wprawy coraz to lepiej połyka, nie może jej jednak strawić;

i nigdy nie nauczy się samodzielnie swych potraw przyrządzać”. W książce tej autor zwraca również uwagę na problem przywiązywania przez uczniów nadmiernej wagi do otrzymywanych ocen (zob. [7], s. 96): „Uczeń widzi w nauce nie wartości istotne a tylko możliwość uzyskania tych czy innych stopni, za którymi idzie patent. Ten świstek papieru, któremu ustawa nadała moc prawną, jest wytyczną jego działania. Z nielicznymi wyjątkami dziecko nie chce posiadać wiedzy [...]. W zasadzie uczniowi chodzi o dodatni stopień i za niego, pobawmy się w paradoks, gotów zrzec się całej wiedzy”. Autor stawia dalej tezę, że w związku z tym jako jeden z najważniejszych środków nauki uczeń widzi ściąganie, zaś: „To absurdalne stanowisko jest naturalnym wynikiem utożsamienia nauki z oceną, uzależnieniem wartości intelektualnych od praktycznych celów”. Dodaje też, że: „Można zapatrywanie to zwalczać, jeżeli ktoś lubi donkiszoterię i walkę z wiatrakami, ale jest ono racjonalne z punktu widzenia dzieci, a ojcami jego jesteśmy my sami, pedagogowie, którzy oddzieliliśmy się od dusz dziecięcych murem naszego systemu not i kar” (zob. [7], s. 96).

Biorąc pod uwagę powyższe rozważania, można stwierdzić zatem, że w procesie skutecznego uczenia kluczowe jest tzw. ocenianie kształtujące (ang. *formative assessment*). Termin ten wprowadził do edukacji w latach 70. ubiegłego stulecia Benjamin Samuel Bloom [1], wskazując jako główną cechę oceniania kształtującego informację zwrotną, która ma pomóc uczniowi się uczyć oraz umożliwić nauczycielowi doskonalenie techniki nauczania. Pod koniec lat 90. ubiegłego wieku Brytyjskie Stowarzyszenie Badań Edukacyjnych powołało specjalny zespół pracujący nad szkolnym ocenianiem, zaś po roku 2000 ocenianie kształtujące stało się ważnym przedmiotem badań naukowych. Prof. John Hatti z Nowej Zelandii, podsumowując w 2009 roku wyniki 1287 badań edukacyjnych, stwierdził, iż wpływ informacji zwrotnej przekazywanej uczniowi jest silniejszy niż inne możliwe do zastosowania sposoby oddziaływania nauczyciela na ucznia [3].

W książce *Uczę (się) w szkole* Danuta Sterna stwierdza (zob. [8], s. 17): „w ocenianiu kształtującym staramy się, aby uczniowie byli gotowi wziąć odpowiedzialność za swój proces uczenia się, aby bardziej zależało im na wiedzy niż na wynikach testów, które tę wiedzę sprawdzają”. Opisuje ona pięć podstawowych strategii oceniania kształtującego, które można streścić w poniższych punktach (zob. [8], s. 18–19):

1) **„Określanie i wyjaśnianie uczniom celów uczenia się i kryteriów sukcesu.**

Z perspektywy ucznia: Lepiej się uczyć, jeśli wiem po co i czego mam się nauczyć.

2) **Organizowanie w klasie dyskusji, zadawanie pytań i zadań dających informacje, czy i jak uczniowie się uczą.**

Z perspektywy ucznia: Lepiej się uczyć, jeśli nauczyciel rozmawia ze mną o moich postępach i w każdej chwili wie, na jakim etapie nauki jestem.

3) **Udzielanie uczniom takich informacji zwrotnych, które umożliwiają im widoczny postęp.**

Z perspektywy ucznia: Lepiej się uczyć, jeśli nauczyciel udziela mi informacji zwrotnej, co zrobiłem dobrze, co i jak powinienem poprawić i jak mogę się dalej rozwijać.

4) **Umożliwianie uczniom korzystania z siebie nawzajem jako zasobów edukacyjnych.**

Z perspektywy ucznia: Lepiej się uczyć, jeśli korzystam z wiedzy i umiejętności moich koleżanek i kolegów.

5) **Wspomaganie uczniów, by stali się odpowiedzialnymi autorami procesu swojego uczenia się.**

Z perspektywy ucznia: Lepiej się uczyć, gdy jestem świadomy, jak przebiega proces mojego uczenia się i odpowiadam za niego”.

Opisaną wyżej koncepcję oceniania kształtującego można z powodzeniem stosować w procesie kształcenia studentów wyższych uczelni podczas zajęć laboratoryjnych. W odróżnieniu od wykładów, gdzie głównym celem jest jedynie przekazywanie określonych treści, zajęcia laboratoryjne umożliwiają bardziej indywidualne podejście do studenta, ponieważ prowadzone są w małych grupach. W trakcie tego typu zajęć prowadzący ma możliwość bezpośredniej weryfikacji posiadanych przez studenta umiejętności. Zakłada się przy tym, że student nie tylko ma rozwiązywać kolejno stawiane przed nim zadania, ale należy wymagać od niego również dokonywania jasnej prezentacji tych rozwiązań, co będzie stanowić dowód zrozumienia materiału. Poprzez wskazywanie niedociągnięć oraz porównanie oceny zaprezentowanego rozwiązania przez innych, student ma możliwość rozwijania swoich umiejętności praktycznych jak również samodzielnej oceny swoich dokonań.

Na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej dla studentów kierunku informatyka od lat prowadzone są zajęcia z przedmiotu podstawy informatyki. W ramach tych zajęć prowadzone są wykłady, ćwiczenia tablicowe oraz ćwiczenia laboratoryjne. Podczas zajęć laboratoryjnych studenci zapoznają się między innymi z zagadnieniami dotyczącymi budowy i działania komputerów o architekturze von Neumanna na bardzo prostym przykładzie maszyny W – komputera wykładowego zaprojektowanego w latach siedemdziesiątych ubiegłego wieku przez prof. Stefana Węgrzyna [9]. Początkowo studenci poznawali zasady działania komputera przy użyciu fizycznej implementacji projektu [2], obecnie zaś do tego celu wykorzystywany jest programowy symulator maszyny W.

Niniejszy artykuł jest próbą pokazania, jak mogą wyglądać przykładowe zajęcia laboratoryjne. Podczas tych zajęć studenci będą rozwiązywać kolejne zadania, poznając podstawy działania maszyny W w praktyce: rozpoczynając od uświadomienia sobie przeznaczenia i roli poszczególnych rejestrów oraz układu sterującego, poprzez projektowanie przesyłów międzyrejestrowych i rozkazów dla procesora, aż po użycie uprzednio zaprojektowanych rozkazów w krótkim programie testującym poprawność ich działania. Zakłada się przy tym, że zajęcia te realizowane są w ramach jednego spotkania trwającego 3 godziny zegarowe (czyli 4 godziny lekcyjne, jak ma to miejsce obecnie).

Przy tworzeniu zaprezentowanego niżej konspektu wykorzystywano sugestie zawarte w książce *Technologie informacyjno-komunikacyjne na lekcjach. Przykładowe konspekty i polecane praktyki* [6]. Autorki przedstawiają tam wskazówki jak tworzyć konspekty lekcji z zastosowaniem oceniania kształtującego, sugerują stawianie poleceń pobudzających głębokie myślenie uczniów oraz zadawanie pytań otwartych, dając przy tym uczniom czas na ustalenia odpowiedzi. Podkreślają przy tym, iż należy wykorzystywać w procesie uczenia błędne odpowiedzi uczniów i zwracać uwagę na fakt, że informacja zwrotna wiąże się z kryteriami sukcesu (zob. [6], s. 24): „określa, co jest dobre, co wymaga poprawy, w jaki sposób tę poprawę wykonać i jak postępować dalej”, a podstawą uczenia się jest szukanie i eliminowanie błędów.

2. Konspekt ćwiczeń laboratoryjnych

Zanim przejdziemy do opisu planowanego przebiegu zajęć, warto określić formalnie kilka podstawowych faktów.

- Temat: *Projektowanie rozkazów dla maszyny W.*
- Cel: praktyczne poznanie zasad działania układu sterującego komputera w architekturze von Neumanna.
- Kryteria sukcesu dla studenta:

- umiem opisać i scharakteryzować najważniejsze elementy składowe komputera w architekturze von Neumanna,
 - rozumiem zapisy przesyłów międzyrejestrowych w notacji nawiasowej,
 - umiem zaprojektować przesył międzyrejestrowy dla maszyny W,
 - umiem sprawdzić działanie napisanego przesyłu w symulatorze maszyny W,
 - rozróżniam pojęcie przesyłu międzyrejestrowego od rozkazu procesora,
 - umiem zaprojektować nowy rozkaz dla maszyny W,
 - umiem przetestować działanie napisanego rozkazu w symulatorze maszyny W.
- Dotychczasowa wiedza i umiejętności uczniów:
zakłada się, że student poznał na wykładzie z przedmiotu podstawy informatyki podstawowe elementy konstrukcyjne komputera, takie jak: rejestry, liczniki, jednostka arytmetyczno–logiczna, układ pamięci, układ sterujący; umie wyjaśnić różnice między poziomowymi i impulsowymi sygnałami sterującymi; zna pojęcia cyklu zegarowego (taktu) oraz cyklu instrukcji; potrafi opisać schemat budowy i zasady działania komputera w architekturze von Neumanna na przykładzie maszyny W; zna standardową listę rozkazów maszyny W.

Ponadto zakłada się, że podczas ćwiczeń tablicowych student zapoznał się z notacją nawiasową służącą do opisu przesyłów międzyrejestrowych oraz potrafi teoretycznie zaprojektować proste przesyły i rozkazy dla maszyny W [4].

- Wykorzystywane narzędzia TIK¹ oraz cel ich zastosowania:
podczas zajęć studenci korzystać będą z programowego symulatora maszyny W², aby lepiej zrozumieć działanie komputera, mając możliwość bezpośredniego śledzenia zmian zawartości poszczególnych rejestrów oraz pamięci operacyjnej w zależności od aktywowanego sygnału sterującego.

Poniżej przedstawiony zostanie przykładowy przebieg zajęć laboratoryjnych. Ze względu na przejrzystość zastosowano tu podział na trzy etapy. Najpierw studenci podczas dyskusji z prowadzącym powinni przypomnieć sobie, jak zbudowana jest maszyna W i jakie są konsekwencje zastosowania architektury von Neumanna. Następnie studenci, pracując samodzielnie lub w parach, realizują pierwsze zadania – projektują przesyły międzyrejestrowe przy nieaktywnym układzie sterującym jednostki centralnej. Na ostatnim etapie studenci projektują rozkazy, zapisują je w odpowiedniej dla symulatora maszyny W notacji, a następnie testują ich działanie, pisząc krótkie programy z użyciem utworzonych rozkazów. Każdy z tych etapów jest opisany w oddzielnym podrozdziale.

2.1. Wprowadzenie – przypomnienie charakterystyki maszyny W

Na początku zajęć prowadzący prosi studentów o uruchomienie symulatora maszyny W (rys. 1), następnie pyta studentów o poszczególne elementy składowe jednostki centralnej.

- Do czego służy rejestr I (instrukcji)?
Dlaczego jest on podzielony na dwie części?
Dlaczego ma on wejście od strony magistrali danych?
Dlaczego ma on wyjście na magistralę adresową?

¹ Skrót TIK jest używany dla opisu technologii informacyjno–komunikacyjnych.

² Jest on dostępny na Platformie Zdalnej Edukacji Politechniki Śląskiej dla studentów zapisanych na kurs Podstawy Informatyki.

lewym przyciskiem myszy, studenci realizują wykonanie taktu (cyklu zegarowego), wybierając odpowiednią opcję z menu *wykonaj* lub przy użyciu przycisku F7. Warto przy tym zwrócić uwagę, że oba sygnały (*wyl*, *wea*) muszą być aktywne równocześnie w tym samym takcie, gdyż aktywacja tylko jednego sygnału sterującego w pojedynczym takcie spowoduje powstanie błędu. Gdybyśmy chcieli w jednym takcie aktywować jedynie sygnał *wyl*, a w kolejnym sygnał *wea*, przy próbie wprowadzenia wartości do rejestru A symulator maszyny W zgłosi komunikat o błędzie: *Magistrala A jest pusta*.

W tym momencie warto przypomnieć studentom i zwrócić uwagę na konsekwencje stosowania poziomowych i impulsowych sygnałów sterujących. Sygnał *wyl* jest poziomowy, czyli jest aktywny przez cały czas trwania cyklu zegarowego (taktu) i jest używany przy wyprowadzaniu zawartości rejestru na magistralę. Natomiast sygnał *wea* jest impulsowy, czyli jest aktywowany na końcu cyklu zegarowego (taktu) i służy do wprowadzania wartości z magistrali do rejestru.

Następnie prowadzący zadaje studentom pytanie, co spowoduje równoczesna aktywacja sygnałów *wyl*, *wea*, *il* (sygnały aktywujemy w tym samym cyklu zegarowym), a po chwili studenci proszeni są o sprawdzenie swojej odpowiedzi przy użyciu symulatora maszyny W. Warto również przypomnieć, co się stanie, gdy do aktywnych w jednym takcie sygnałów *wyl*, *wea*, *il* dołączymy dodatkowo sygnał *czyt*. Aby lepiej zobrazować tę sytuację, prowadzący może poprosić studentów o wypełnienie kilku kolejnych komórek pamięci operacyjnej różnymi wartościami i śledzenie zawartości rejestrów L, A oraz S podczas kilkukrotnego wykonywania taktu (F7).

Przeprowadzone eksperymenty prowadzący może podsumować uwagą, aby optymalizację czasową realizacji przesyłów międzyrejestrowych dokonywać z odpowiednim wyczuciem i troską o poprawne wykonanie zadania, za każdym razem porównując otrzymane w symulatorze maszyny W wyniki z wartościami przewidywanymi. Należy przy tym uważnie czytać zgłaszane przez symulator komunikaty. Jako przykład prowadzący może wskazać próbę jednoczesnej aktywacji sygnałów *wyl*, *wyad* (próba wysłania na magistralę adresową dwóch wartości: z rejestru licznika i z rejestru instrukcji). Wykonanie taktu w tej sytuacji skutkuje komunikatem o błędzie: *Magistrala A jest już zajęta*.

Następnie warto przypomnieć stosowane w maszynie W kodowanie liczb ujemnych w postaci zapisu uzupełnieniowego do 2. Zastosowanie tego kodowania w architekturze 8-bitowej pozwala na zapisanie liczb z zakresu od -128 do 127 . Zauważamy, że wartość 127 po zinkrementowaniu zmieni się na -128 . Prowadzący może poprosić studentów o wprowadzenie do rejestru Ak (akumulatora) wartości np. 250 i zapytać ich jaka jest to liczba: dodatnia czy ujemna. Prowadzący zwraca uwagę, że wskazując prawym przyciskiem myszy dowolny rejestr w symulatorze maszyny W, można zmienić domyślny tryb wyświetlania wartości (używając opcji *Pokaż jako* możemy wybrać spośród: *dziesiętnie bez znaku*, *dziesiętnie ze znakiem*, *binarnie*, *szesnastkowo*). Wybierając binarny tryb wyświetlania wartości, widzimy, że w przypadku liczb 8-bitowych dla wartości 250 pierwszy od lewej bit (bit znaku) jest równy jeden. Zatem wprowadzona do akumulatora wartość jest ujemna (w architekturze 8-bitowej).

Na koniec pierwszej części zajęć prowadzący prosi studentów o realizację przesyłu (Ak) \rightarrow L , czyli o przesłanie zawartości rejestru Ak (wartości 250) do rejestru L. Dla domyślnej, standardowej architektury maszyny W (z uwagi na fakt, że wyjście Ak jest podpięte do magistrali danych, zaś wejście L do magistrali adresowej) jedynym możliwym rozwiązaniem jest przesył za pośrednictwem rejestru instrukcji. Zatem w jednym takcie należy uaktywnić sygnały *wyak*, *wei*, w następnym zaś *wyad*, *wel*. Warto przy tym zwrócić uwagę, jaka wartość pojawi się docelowo w rejestrze L. W rejestrze tym powinna pojawić się wartość 26 (binarnie 11010), gdyż sygnał *wyad* umożliwia wyprowadzenie na magistralę adresową jedynie 5 bitów z 8-bitowego słowa (3 najstarsze bity słowa zostają „obcięte”).

W tym momencie prowadzący przypomina studentom o możliwości rozszerzenia architektury maszyny W o połączenie międzymagistralowe (menu *Projekt* → *Opcje..* → *Składniki* → *W+*), prosi o jednoczesną aktywację sygnałów *wyak*, *sa*, *wel* i wykonanie taktu (F7), po czym zadaje pytanie, co się stało i dlaczego. Zawartość rejestru licznika nie zmieniła się, gdyż w standardowej 8-bitowej architekturze maszyny W magistrala adresowa jest węższa od magistrali danych i można nią przesyłać jedynie 5-bitowe wartości. Zatem nawet przy wykorzystaniu połączenia międzymagistralowego maszyny W+ nie jest możliwe przesłanie 8 bitów słowa z 8-bitowej magistrali danych na 5-bitową magistralę adresową.

2.2. Projektowanie przesyłów międzyrejestrowych

Drugą część zajęć stanowi praca własna studentów realizujących podane przez prowadzącego przykładowe przesyły międzyrejestrowe. Każdemu ze studentów prowadzący zadaje do zaprojektowania dwa przesyły. Jeden z nich jest prostym, kilkuktowym przesyłem, możliwym do wykonania w architekturze maszyny W. Drugi z nich jest bardziej skomplikowany i wymaga rozszerzonej architektury W+.

Kilka przykładowych propozycji przesyłów to:

- 1) $((Ad)) \rightarrow Ak$ oraz $(Ak) - 1 \rightarrow 0$,
- 2) $((L)) \rightarrow Ak$ oraz $|((Ak))| \rightarrow (2)$,
- 3) $(Ak) \rightarrow (L)$ oraz $((Ad) + 1) - ((Ad)) \rightarrow Ak$,
- 4) $(Ak) - ((L)) \rightarrow Ak$ oraz $(Ad) - (Ak) \rightarrow 1$,
- 5) $(Ak) \rightarrow (Ad)$ oraz $((Ad)) + ((Ad) + 1) \rightarrow 0$,
- 6) $3 * (Ak) \rightarrow Ak$ oraz $(0) + (1) \rightarrow (Ad)$,
- 7) $0 \rightarrow (Ad)$ oraz $|((0))| \rightarrow (1)$,
- 8) $(Ak) \rightarrow 0$ oraz $((Ad)) + 1 \rightarrow (0)$.

Po podaniu zadań warto przedyskutować ze studentami kwestię interpretacji treści (zadania podawane są w notacji nawiasowej). Ważne jest, aby każdy z uczestników laboratorium dobrze zrozumiał treść swojego zadania. Prowadzący może zapytać wybranych studentów o treść konkretnych zadań, podając przykłady jak poniżej.

- Co oznacza zapis $(Ak) \rightarrow S$?
- Jak należy interpretować zapisy $(Ad) \rightarrow Ak$, $((Ad)) \rightarrow Ak$ oraz $((((Ad)))) \rightarrow Ak$?
- Jak należy interpretować zapisy $(Ak) \rightarrow (Ad)$ oraz $(Ak) \rightarrow ((Ad))$?
- Co stanowi błąd w zapisie $(Ak) \rightarrow Ad$?
- Dlaczego zapis $Ak \rightarrow Ad$ jest niepoprawny?

Prowadzący zwraca również uwagę studentów, że w treściach zadań czasami, prócz nazw rejestrów, pojawiają się konkretne wartości liczbowe. Prowadzący przypomina, że zawartości poszczególnych rejestrów nie są określone, zatem odpowiednie wartości liczbowe należy „wypracować” (nie można ręcznie wpisywać potrzebnych wartości do rejestrów maszyny W). Przy tej okazji można przypomnieć, jak uzyskać na przykład wartość

- zero, jako realizację przesyłu $(Ak) - (Ak) \rightarrow Ak$;
- jeden, jako realizację przesyłu $0 \rightarrow L$, a następnie $(L) + 1 \rightarrow L$.

Studenci powinni też pamiętać o możliwości skorzystania z różnicy kolejnych wartości licznika, przed i po jego inkrementacji.

W kontekście pojawiających się liczb w zapisie nawiasowym warto też zapytać o interpretację poniższych zapisów.

- Co oznaczają zapisy: $1 \rightarrow Ak$, $(1) \rightarrow Ak$, $((1)) \rightarrow Ak$?
- Co oznaczają zapisy: $(Ak) \rightarrow 1$, $(Ak) \rightarrow (1)$, $(Ak) \rightarrow ((1))$?

Studenci proszeni są o samodzielną pracę przy projektowaniu przesyłów międzyrejstrowych z użyciem **sterowania ręcznego** w symulatorze maszyny W, natomiast prowadzący, podchodząc do kolejnych stanowisk laboratoryjnych, sprawdza wyniki ich prac. Każdy ze studentów powinien pokazać prowadzącemu swój projekt i potrafić uzasadnić wybór aktywacji konkretnych sygnałów sterujących w poszczególnych taktach rozwiązania.

Jeśli studenci zakończyli już pracę nad swoim projektem i czekają na ocenę ich rozwiązania przez prowadzącego, powinni zostać zachęcani do skonsultowania swoich rozwiązań z innymi oczekującymi. Prowadzący może również zachęcić studentów do cichej dyskusji w małych grupach i dzielenia się swoimi spostrzeżeniami z innymi. Można też zasugerować oczekującym studentom, aby zaczęli przygotowywać się do kolejnej części laboratorium, której celem jest projektowanie rozkazów (studenci mogą poznać wcześniej treści swoich zadań).

2.3. Projektowanie rozkazów oraz testowanie ich działania

Trzecią i ostatnią część zajęć laboratoryjnych stanowi praca własna studentów realizujących podane przez prowadzącego przykładowe rozkazy dla maszyny W lub rozszerzonej architektury W+. Na tym etapie zakłada się, że studenci potrafią już samodzielnie zaprojektować przesyły międzyrejstrowe stanowiące treść rozkazów. Warto przypomnieć studentom, że podczas projektowania rozkazu prócz zaprojektowania przesyłu trzeba też pamiętać o poprawnym wykonaniu całego cyklu rozkazowego („pobierz, zdekoduj, wykonaj”). Można przy tym zdefiniować rozkaz bezczynności NOP (ang. *no-operation*), który nic nie robi. Musi on zawierać dwa takty:

- 1) *czyt wys wei il* – realizujący odczytanie treści rozkazu z pamięci operacyjnej, przesłanie go i zdekodowanie w rejestrze instrukcji oraz równocześnie przygotowujący adres kolejnej instrukcji w rejestrze licznika,
- 2) *wyl wea* – realizujący przesłanie adresu kolejnej instrukcji do rejestru adresowego pamięci operacyjnej tak, aby w kolejnym takcie można było poprawnie ją odczytać (rozpocząć kolejny cykl instrukcji).

Podkreślamy przy tym, że rejestr instrukcji i rejestr licznika są bardzo wrażliwe na zmiany ze względu na swoje funkcje. W trakcie całego cyklu rozkazowego można użyć tylko jeden raz sygnału *wei* (sygnał ten powinien być aktywowany tylko w pierwszym takcie³) oraz trzeba pamiętać, aby w ostatniej fazie rozkazu rejestry L i A miały tę samą zawartość – adres kolejnej instrukcji.

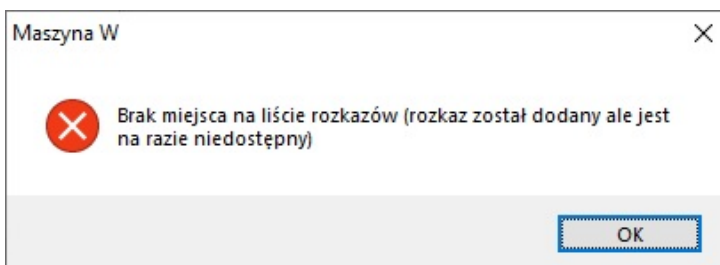
³ Każdorazowe użycie sygnału *wei* powoduje na nowo dekodowanie rozkazu, którego kod znajduje się w części kodowej rejestru instrukcji (a wtedy jego właściwa realizacja rozpoczyna się w kolejnym takcie).

Warto przy tym wraz ze studentami omówić kolejne takty kilku wybranych rozkazów ze standardowej listy rozkazów maszyny W (widocznej wybierając menu *Widok* → *Lista rozkazów* lub przy użyciu przycisku F10), zwracając uwagę zarówno na sposób podawania treści rozkazu (aktywacja sygnałów sterujących, średniki kończące takty), jak i słowa kluczowe typu ROZKAZ, ARGUMENTY, KONIEC, JEŻELI itp. Na przykładzie rozkazu SOM warto prześledzić możliwość realizacji rozkazu warunkowego.

Dobrze jest pokazać studentom, jak wprowadzić kolejny rozkaz na listę rozkazów, wybierając menu *Plik* → *Nowy* → *Rozkaz* i wpisując w nowe okno następującą treść:

```
ROZKAZ NOP;  
ARGUMENTY 0;  
czyt wys wei il;  
wyl wea;
```

Zwracamy uwagę na pojawiający się podczas kompilacji (prawy przycisk myszy *Kompiluj* lub CTRL+F9) komunikat widoczny na rysunku 2, który informuje, że rozkaz jest już zdefiniowany na liście rozkazów, ale nie jest na razie dostępny. Można zapytać studentów jak rozumieją ten komunikat i zasugerować im poprawną odpowiedź. W tym celu przypominamy, ile rozkazów do tej pory widnieje na liście rozkazów i porównujemy tę liczbę z liczbą możliwych stanów, które można zakodować na 3 bitach (w domyślnej architekturze maszyny W 3 bity służą do kodowania rozkazu, a 5 bitów do kodowania adresu). Zatem rozwiązanie problemu niedostępności nowego rozkazu polega na rozszerzeniu standardowej architektury maszyny W o dodatkowy bit służący do kodowania rozkazu (menu *Projekt* → *Opcje...* → *Architektura*). Warto przy tym podkreślić, że ten dodatkowy bit pozwala na poprawne dodanie 8 nowych rozkazów.



Rysunek 2. Komunikat o błędzie

Na koniec części wstępnej, zanim studenci zajmą się pracą samodzielną, warto omówić jeszcze w jaki sposób można testować działanie rozkazów. Można to zrobić na dwa alternatywne sposoby:

- 1) poprzez „obliczenie” kodów rozkazów i wprowadzenie ręczne pożądaných wartości do odpowiednich komórek pamięci operacyjnej
- 2) lub za pomocą krótkiego programu napisanego w języku assemblera maszyny W (menu *Plik* → *Nowy* → *Program*) i jego kompilacji (prawy przycisk myszy *Kompiluj* lub CTRL+F9), co powoduje wypełnienie zawartości pamięci operacyjnej odpowiednimi wartościami.

Można pokazać na przykład program testujący rozkaz SOM (skok pod zadany adres, o ile zawartość akumulatora jest ujemna):

```
POB A  
SOM B  
DOD A
```

B: STP
A: RST 10

Warto zasugerować studentom śledzenie zawartości rejestrów maszyny podczas wykonywania tego programu w trybie pracy krokowej⁴. Mamy przy tym dwie możliwości: wykonywanie programu rozkaz po rozkazie (menu *Wykonaj* → *Rozkaz* lub F8) lub takt po takcie (menu *Wykonaj* → *Takt* lub F7). Po zakończeniu działania naszego programu testowego w akumulatorze powinniśmy widzieć wartość 20 (podwojenie wartości zapisanej w komórce pamięci o adresie symbolicznym A). Można teraz zmienić zawartość komórki A, wpisując zamiast RST 10, np. wartość RST -10 i zobaczyć, że po skompilowaniu i ponownym uruchomieniu programu, po zakończeniu działania programu, w akumulatorze zostaje wartość -10. Jednak sposób prezentacji tej liczby zależy od wybranego trybu wyświetlania zawartości rejestru Ak i możemy zobaczyć -10 (gdy wyświetlamy liczbę ze znakiem) lub 502 (w przypadku liczby bez znaku), lub 111110110 (binarnie).

Należy też przypomnieć studentom o możliwości realizacji rozkazów dwuargumentowych (przyjmujemy wtedy, że następny rozkaz znajduje się 2 komórki dalej za projektowanym rozkazem, natomiast komórka następna służy do przekazania drugiego argumentu rozkazu). Wymagana jest wówczas dodatkowa inkrementacja zawartości licznika. Z kolei w przypadku rozkazów realizowanych z zastosowaniem adresacji pośredniej bardzo ważne jest odpowiednie zaplanowanie treści programu testującego tak, aby odpowiednio zaprezentować działanie rozkazu. W przypadku testowania rozkazów warunkowych należy tak zaprojektować program testowy, aby można było sprawdzić działanie obu wersji rozkazu. Warto też zwrócić uwagę, że w wielu przypadkach do realizacji operacji arytmetycznych wykorzystywany jest akumulator, więc (jeśli nawet w treści zadania nie wspomina się wprost o odtwarzaniu jego wartości początkowej) o ile to możliwe należy zapamiętywać i przywracać jego poprzednią wartość.

Podobnie jak uprzednio, podczas gdy studenci zajęci są samodzielną pracą przy projektowaniu rozkazów i testowaniu ich poprawności, prowadzący podchodzi do kolejnych stanowisk laboratoryjnych i sprawdza wyniki prac studentów. Każdy ze studentów powinien umieć przedstawić projekt rozkazu w postaci pliku rozkazu oraz uruchomić program testujący jego działanie. Prowadzący powinien zwracać uwagę nie tylko na poprawność realizacji rozkazów, ale również na optymalizację czasową ich działania.

Każdy ze studentów dostaje do zaprojektowania dwa rozkazy: jeden prostszy (dla architektury W) i jeden trudniejszy (dla rozszerzonej architektury W+). Przykładowe rozkazy podane są poniżej:

- 1) NEG: $-(Ak) \rightarrow Ak$ oraz DDP Ad: $(Ak) + (((Ad))) \rightarrow Ak$,
- 2) NEG Ad: $-((Ad)) \rightarrow (Ad)$ oraz ŁDP Ad: $(Ak) \rightarrow ((Ad))$,
- 3) MIN Ad: $\min\{(Ak), ((Ad))\} \rightarrow Ak$ oraz INC: $(Ak) + 1 \rightarrow Ak$,
- 4) MAX Ad: $\max\{(Ak), ((Ad))\} \rightarrow Ak$ oraz INC Ad: $((Ad)) + 1 \rightarrow (Ad)$,
- 5) ABS Ad: $|((Ad))| \rightarrow Ak$ oraz DEC Ad: $((Ad)) - 1 \rightarrow (Ad)$,
- 6) PGM Ad: jeżeli $(Ak) < ((Ad))$, to $(Ad) \rightarrow L$ oraz DEC: $(Ak) - 1 \rightarrow Ak$,
- 7) DD2 Ad (L)+1: $((Ad)) + ((L) + 1) \rightarrow Ak$ oraz PBP Ad: $((((Ad))) \rightarrow Ak$,
- 8) MN3 Ad: $3 * ((Ad)) \rightarrow Ak$ oraz ŁD2 Ad (L)+1: $((Ad)) \rightarrow ((L) + 1)$.

⁴ Należy przypomnieć studentom o potrzebie wyłączenia opcji **sterowanie ręczne** w widoku jednostki centralnej.

2.4. Podsumowanie

Zwróćmy uwagę, że nie zawsze studenci są odpowiednio przygotowani do zajęć, a czasami po prostu w wolniejszym tempie przyswajają wiedzę. Można zatem zasugerować osobom, które nie były w stanie dokończyć realizacji wszystkich zadań w regulaminowym czasie (3 godziny zegarowe), napisanie sprawozdania zawierającego rozwiązania wszystkich zadań wraz z odpowiednimi komentarzami i dodatkowy kontakt mailowy lub osobisty w czasie konsultacji. Można też zalecić studentom, którzy nie czują się zbyt pewni w rozwiązywaniu prezentowanych zadań, aby w ramach zadania domowego spróbowali zrealizować pozostałe zadania z listy zadań – takie, które nie zostały im przydzielone podczas zajęć laboratoryjnych. Warto przypominać studentom, że studiowanie wymaga od nich samodzielności w większym stopniu niż miało to miejsce w szkole średniej oraz aby nie zapominali o możliwości regularnych konsultacji z prowadzącymi tak, aby w pełni wykorzystać ich wiedzę w procesie nabywania nowych umiejętności.

Pamiętajmy też o zaleceniach opisanych w książce *Cele uczenia się. Jak pomóc uczniom zrozumieć każdą lekcję*. Autorki książki stwierdzają, że należy zadawać uczniom zadanie domowe, które jednak „nie powinno nigdy wymagać nauczania się czegoś nowego albo wykonania ćwiczenia, które nie będzie jasne [...]”. Jeśli rozwiązanie zadania domowego przyjdzie z dużym trudem, oznacza to, że utrwalone zostaną nieścisłości i braki w wiedzy. Uczniowie powinni uczyć się nowych pojęć i procesów podczas bieżącej lekcji, we współpracy z nauczycielem. Zadanie domowe może jednak być efektywne i zostać wykorzystane w związku z celem uczenia się, jeśli poprosimy uczniów o wykonanie w domu pracy, która poszerza to, czego dokonaliśmy na lekcji, aby opanować dany materiał” (zob. [5], s. 68). W podobnym tonie wypowiadają się autorki książki *Technologie informacyjno-komunikacyjne na lekcjach. Przykładowe konspekty i polecane praktyki*, które stwierdzają ponadto, że „praca domowa powinna mieć sens, być jasna i atrakcyjna; powinna też dawać wybór i zostać bezzwłocznie sprawdzona” (zob. [6], s. 26).

Warto też pod koniec zajęć zadać studentom pytanie, w jakim stopniu przeprowadzone zajęcia pomogły im w osiągnięciu sukcesu zdefiniowanego na wstępie i zasugerować im zastanowienie się, co było pomocą, a co przeszkodą w zdobywaniu nowych umiejętności. Warto wysłuchać ich uwag tak, aby sam prowadzący w przyszłości mógł udoskonalać swój sposób przekazywania i weryfikowania wiedzy.

Literatura

1. B.S. Bloom, J.T. Hastings, G.F. Madaus, *Handbook of formative and summative evaluation of student learning*, McGraw-Hill, New York 1971.
2. M. Chłopek, R. Tutajewicz, *Ćwiczenia laboratoryjne z Podstaw Informatyki – maszyna W*, skrypt uczelniany Politechniki Śląskiej nr 2062, Wydawnictwo Politechniki Śląskiej, Gliwice 1997.
3. J. Hattie, *Visible learning: a synthesis of over 800 meta-analyses relating to achievement*, Routledge, New York 2009.
4. A. Momot, R. Tutajewicz, *Maszyna W – jak zaprojektować prosty rozkaz*, MINUT 2019 (1), s. 24–35.
5. C.M. Moss, S.M. Brookhart, *Cele uczenia się. Jak pomóc uczniom zrozumieć każdą lekcję*, Centrum Edukacji Obywatelskiej, Warszawa 2014.
6. M. Ostrowska, D. Sterna, *Technologie informacyjno-komunikacyjne na lekcjach. Przykładowe konspekty i polecane praktyki*, Centrum Edukacji Obywatelskiej, Warszawa 2015.
7. I. Schreiber, *Ściąga w praktyce szkolnej*, Księgarnia Nauka i Sztuka, Kraków 1938.
8. D. Sterna, *Uczę (się) w szkole*, Centrum Edukacji Obywatelskiej, Warszawa 2014.
9. S. Węgrzyn, *Podstawy informatyki*, PWN, Warszawa, 1982.