

1. Co to jest procesor?

Procesor to układ elektroniczny, który działa w sposób programowy, co polega na samodzielnym pobieraniu i wykonywaniu rozkazów umieszczonych w jego pamięci programu.

2. Kiedy działanie procesora może być przerwane?

Działanie procesora może być przerwane przez tzw. stany specjalne procesora, które zawieszają realizację programu przez procesor. Stany specjalne to:

- stan wstrzymania – występuje przy trybie DMA, po przyjęciu sygnału HOLD procesor kończy realizowany cykl maszynowy i ulega swoistemu zamrożeniu, tj. nie wykonuje żadnych czynności i wprowadza stany wysokiej impedancji na szynę adresową i szynę danych
- stan oczekiwania – wprowadzany w niego w trakcie trwania cyklu maszynowego za pomocą odpowiedniego sygnału (w Z80 – WAIT) na tak długo jak trwa ten sygnał, nie odcina procesora od magistrali.
- stan zatrzymania – wprowadzany w sposób programowy przez rozkaz (w Z80 HALT). Procesor przestaje wykonywać rozkazy i oczekuje na przyjęcie sygnału przerwania. Po wykonaniu podprogramu obsługi przerwania kontynuuje prace od rozkazu następnego po rozkazie HALT.

3. Co to znaczy, że działa w sposób programowy?

Polega to na samodzielnym pobieraniu i wykonywaniu rozkazów umieszczonych w jego pamięci (procesora). Oznacza to, że wszystko co robi nadzorowane jest przez program.

4. Co to jest rozkaz? Gdzie składujemy podczas wykonywania?

Rozkaz to elementarne polecenie dla procesora. Ciąg rozkazów tworzy program. Rozkazy mają postać liczb binarnych, bo tylko takie polecenia są zrozumiałe dla procesora. Aby procesor mógł samodzielnie pobrać i wykonać rozkazy, muszą być umieszczone w pamięci programu, czyli w takich obszarach pamięci, które procesor widzi jako zawierające program. Rozkaz dzieli się na kod rozkazu i może zawierać także argument lub argumenty. Rozkaz musi określać wszystkie czynności wykonywane przez procesor podczas realizacji tego rozkazu.

5. Co to jest komenda?

Pewnie Kamil coś jebnął bo nie ma tego ani w słowniku ani w jego książce :-D

6. Co to jest instrukcja?

Polecenie języka symbolicznego (ciąg symboli) tłumaczone na rozkazy. W assemblerze instrukcja tłumaczona jest na jeden rozkaz (1:1).

7. Co to jest kod rozkazu?

Cześć rozkazu, która mówi procesorowi co ma robić. Zawsze jest to pierwszy bajt rozkazu.

8. Co to jest argument?

Argument to część rozkazu, która zawiera dane niezbędne do jego wykonania. Argument nie musi występować w rozkazie. Argument jest 8-bitowy. Rozkaz może zawierać 1 lub więcej argumentów.

9. Czym może być argument?

Gdy rozkaz zawiera 2 argument, to jest to na ogół argument operacji arytmetycznej lub logicznej, czyli jako taka liczba na której są wykonywane operacje – wtedy jest to argument bezpośredni. Pojedynczy argument rozkazu może być także traktowany jako adres, gdy rozkaz służy do komunikacji procesora z układami I/O o adresach 8-bitowych lub używana jest pamięć danych adresowana 8-bitowo, wtedy jest to adres bezpośredni. Jeśli argument jest 2-bajtowy, to jest traktowany jako adres 16-bitowy bezpośredni, ale każdy z tych bajtów może być traktowany oddzielnie jako 2 argumenty 8-bitowe do innych operacji w bardziej złożonych procesorach obsługujących wielobajtowe rozkazy.

10. Co to jest pamięć?

Pamięć to spójny zbiór komórek pamięci, czyli takich, których adresy stanowią ciągły obszar (adresy dwóch sąsiadujących komórek różnią się o 1).

11. Jak jest zbudowana (pamięć)?

Pamięć zbudowana jest z komórek, które są układami służącymi do przechowywania elementarnej porcji informacji w postaci cyfrowej. Jedna komórka może pamiętać 1 bajt informacji w postaci liczby binarnej, więc pamięć jest organizowana bajtowo. Każda komórka ma swój unikatowy adres. Komórki zawierają spójny obszar adresowy, w którym wszystkie możliwe adresy mają swoje fizyczne reprezentacje w postaci komórek pamięci.

12. Jakie są pamięci?

- Pamięć programu – na ogół ROM
- Pamięć danych
- Pamięć dynamiczna (DRAM)
- Pamięć statyczna (SRAM)
- Pamięć tylko do odczytu (ROM) – nieulotna
- Pamięć operacyjna (na ogół RAM)
- Pamięć typu EEPROM
- Pamięć typu flash

13. Dlaczego pamięć programu jest ROM?

Ponieważ podczas realizacji programu procesor wyłącznie odczytuje pamięć.

14. Co to jest słowo?

Słowo to jednostkowa porcja informacji charakterystyczna dla urządzenia (układu), z reguły wyrażona w bitach lub w bajtach. Proste mikro(procesory i kontrolery) mają 1 bajtowe słowo danych i 2 bajtowe słowo adresowe.

- Słowo danych – liczba bitów przesyłanych jednocześnie przy użyciu szyny danych procesora
- Słowo adresowe – liczba bitów przesyłanych równocześnie przez szynę adresową procesora.

15. Omówić IR.

IR to rejestr rozkazów. Służy do przechowywania kodu rozkazu przez cały jego czas realizacji, lecz w niektórych procesorach RISC rozkaz jest wykonywany w jednym takcie zegara, wtedy rejestr ten nie jest niezbędny. Zawartość rejestru rozkazów jest dekodowana, dzięki czemu uzyskiwane są ciągi impulsów sterujących wysyłanych w drugiej fazie realizacji rozkazu do podzespołów procesora w celu taktowania odpowiednich działań cząstkowych, np. pobierania argumentów rozkazu. Podczas wykonywania rozkazu procesor najpierw pobiera kod rozkazu z pamięci do IR.

16. Jak realizowany jest program?

Program realizowany jest w sposób sekwencyjny, czyli rozkazy są wykonywane jeden po drugim. Program można określić jako zbiór sekwencji rozkazów umieszczonych w określonych obszarach pamięci programu.

17. Do czego jest licznik rozkazów? Opisać zasadę działania.

Licznik rozkazów (PC – Program Counter albo PP – Program Pointer) to wyspecjalizowany rejestr procesora. Podczas wykonywania sekwencji rozkazów, procesor musi najpierw pobrać kod rozkazu, a następnie jego argumenty. W tym celu za każdym razem procesor musi podać adres komórki, w której mieszczą się kolejne elementy rozkazu. Właśnie ten adres wyprowadzany jest z licznika rozkazów. Po pobraniu kodu rozkazu zawartość licznika rozkazów musi wskazywać kolejną komórkę pamięci programu. Po pobraniu kodu rozkazu licznik rozkazów musi zostać zwiększony o 1 (kolejne rozkazy umieszczone są w komórkach o rosnących adresach), tak samo jak po każdym odczytaniu kolejnego bajtu z pamięci programu oraz po pobraniu pierwszego bajtu argumentu. Wniosek: Po każdej komunikacji procesora z pamięcią programu zawartość licznika rozkazów zwiększana jest o 1.

18. Co to jest skok? Jakie są rodzaje skoków?

Skok to zmiana sekwencji wykonywanych rozkazów w efekcie czego procesor przemieszcza się w inny obszar pamięci programu. Rodzaje skoków:

- Skok zwykły – realizowany za pomocą rozkazu JMP, który zawiera 16-bitowy argument będący adresem skoku. Po pobraniu kodu tego rozkazu procesor kopiuje argument rozkazu do licznika rozkazów, dzięki czemu kolejny rozkaz będzie pobierany począwszy od komórki pamięci wskazywanej przez ten argument.
- Skok ze śladem – celem tego skoku jest wywołanie podprogramu. (OPIS PYT. NIŻEJ)
- Skok warunkowy – realizowany o ile jest spełniony warunek wskazany w rozkazie skoku. Gdy tak nie jest procesor pobiera kolejny rozkaz po rozkazie skoku.

19. Opisać skok ze śladem.

Skok ze śladem występuje podczas wywołania podprogramu. Podprogram to sekwencja rozkazów zakończona rozkazem powrotu z podprogramu do kontynuacji programu głównego przez procesor. Podstawowy warunek takiego powrotu jest przechowanie zawartości licznika rozkazów w momencie gdy procesor przechodzi do wywołania podprogramu. Dodatkowo stan procesora (czyli stan wszystkich elementów pamiętających procesora) musi być taki sam jak przed wykonaniem podprogramu, dlatego prócz licznika rozkazów przechowuje się zawartości rejestrów procesora w sposób sprzętowy lub programowy:

- Sprzętowy – zapamiętanie stanów wszystkich rejestrów procesora w fazie realizacji rozkazu wywołującego podprogram. Odtworzenie tych stanów również odbywa się sprzętowo podczas realizacji rozkazu powrotu z podprogramu. Zaleta: krótki czas realizacji i zwalnianie programisty z dopilnowania poprawności realizacji.
- Programowy – wymaga użycia odpowiednich rozkazów. Samoczynnie przechowywany jest tzw. ślad czyli stan licznika rozkazów.

20. Co podczas tego skoku jest przechowywane na stosie?

Na stosie przechowywane są zawartości rejestrów procesora podczas skoku ze śladem.

21. W jaki sposób? Czym różnią się te sposoby?

Proces przechowywania i odtwarzania stanu procesora różni się dla dwóch rodzajów podprogramów:

- zwykły – podprogram wywoływany jest w sposób programowy, tj. za pomocą rozkazu o symbolu CALL z argumentem o postaci adresu, od którego zaczyna się podprogram. Realizacja w 2 krokach: najpierw na stosie jest przechowywana aktualna zawartość licznika rozkazów po czym do licznika rozkazów wprowadzany jest argument rozkazu CALL. Podprogram kończy się rozkazem RET, wtedy ze stosu pobierane są 2 bajty i z powrotem wpisane do licznika rozkazów, są to bajty zawierające adres powrotu do programu głównego.
- podprogram obsługi przerwania – inicjowany w sposób sprzętowy, na sygnał przerwania. Istotną różnicą jest fakt że w przypadku obsługi przerwania procesor używa układu przerwań. Układ ten zawiera przerzutniki i rejestry, których stan musi być odpowiednio przechowany i odtworzony po zakończeniu realizacji podprogramu obsługi przerwania. Do ich przechowania nie używa się stosu, bo odbywa się ono wewnątrz układu przerwań. Żeby powrócić do programu głównego używa się specjalnego rozkazu RETI, który odtwarza stan licznika rozkazów oraz stan układu przerwań.

22. Co to jest stos?

Stos to spójny zbiór komórek pamięci danych zorganizowany według reguły LIFO, czyli ostatni zapisany element musi zostać odczytany jako pierwszy.

23. Co to jest wierzchołek stosu?

Wierzchołek stosu to komórka pamięci danych, która została zapisana na stosie jako ostatnia.

24. Co to jest wskaźnik stosu?

Wskaźnik stosu to specjalny rejestr, który przechowuje adres wierzchołka stosu. Podczas zapisu do stosu procesor musi najpierw zmienić zawartość wskaźnika stosu o 1 w górę, aby wskaźnik wskazywał na wolną komórkę pamięci, a następnie zapisać dane pod tym adresem.

25. Co to jest flaga?

Flaga to bit warunku, czyli specyficzny znacznik ustawiany w stan 1 jeśli wynik operacji arytmetyczno – logicznej wykonanej przez procesor spełnia określony warunek. Podstawowe flagi:

- bit przeniesienia C – jeśli podczas operacji wystąpiło przeniesienie wyniku z 8 na 9 bit.
- bit zera – przyjmuje 1 gdy wynik operacji A-L wynosi 0.
- bit parzystości – 1 gdy liczba bitów o wartości 1 jest parzysta. Wykorzystywana do kontroli poprawności transmisji.
- bit znaku S – przyjmuje wartość równą wartości najbardziej znaczącego bitu wyniku operacji A-L.
- Bit nadmiaru V lub OV – 1, gdy w trakcie operacji na liczbie ze znakiem w kodzie dopełnieniowym do dwóch wynik operacji przekroczył zakres wartości dopuszczalnych dla tego kodu.

Stany flag są wykorzystywane podczas realizacji rozkazów warunkowych.

26. Omówić od wystąpienia przerwania do jego obsłużenia.

Gdy pojawia się sygnał przerwania wygenerowany przez układ I/O, procesor dowiaduje się o nim pod koniec wykonywania aktualnego rozkazu (rozkaz jest niepodzielny, nie może być przerwany). W tym celu procesor próbuje sygnały przerwania tuż przed zakończeniem rozkazu, a po stwierdzeniu, że co najmniej 1 jest aktywny pobiera rozkaz obsługi przerwania z kodera przerwania lub z wewnętrznej pamięci ROM. Licznik rozkazów NIE JEST wtedy zwiększany o 1.

27. Szczegółowo omówić obsługę przerwania.

Najpierw procesor przechowuje na stosie zawartość licznika rozkazów, a następnie prowadzi do niego adres podprogramu obsługi przerwania. Podprogram obsługi przerwania musi kończyć się rozkazem RETI, którego realizacja przywraca stan licznika rozkazów sprzed wywołania tego podprogramu, wtedy możliwe jest kontynuowanie programu głównego. RETI przywraca również stan układu przerwania. Przerwanie programowe polega na takich samych działaniach jak podczas obsługi przerwania inicjowanych sygnałowo, ale przyczyna podjęcia tych działań tkwi w realizacji programu. Pierwsza grupa przerwania obejmuje przerwania, których obsługa jest inicjowana umieszczeniem w programie rozkazu używanego do obsługi przerwania sygnałowych (w Z80 RSTn). Użycie tego pozwala na krokowe uruchamianie programu. Druga grupa przerwania jest inicjowana przez zdarzenia zachodzące podczas realizacji programu, różni się od 1 grupy tym, że końcowym efektem tych zdarzeń jest sygnał przerwania generowany wewnątrz procesora, np. dzielenie przez 0.

28. Skąd podczas przerwania procesor pobiera rozkazy, a skąd nie i dlaczego?

Nie pobiera rozkazów z pamięci programu, ponieważ koncepcja obsługi przerwania oparta jest na sprzętowej, niezależnej od realizacji programu inicjacji obsługi przerwania. Pobierany jest CHYBA z pamięci ROM układu przerwania.

29. Warunek powrotu z podprogramu?

Musi być podany w rozkazie adres dokąd ma powrócić. (Dzięki Szymon)

30. Czym się różni sterownik DMA od procesora?

Sterownik DMA odpowiada za transmisję bloków danych między pamięcią, a urządzeniami I/O. Przesyłanie takich bloków przez procesor jest nieefektywne, gdyż do przesłania każdego bajtu niezbędne jest pobranie kodu odpowiedniego rozkazu, co wydłuża czas transmisji, procesor potrzebuje także pośrednika w postaci akumulatora, dzięki czemu transmisja

realizowana jest w 2 cyklach. Do transmisji używa się sterownika DMA, który umożliwia przesłanie 1 bajtu danych w 1 cyklu.

31. Tryb DMA

Jest to ciąg działań, których celem jest przesłanie bloku danych bez udziału procesora między pamięcią a urządzeniami I/O. Stosowany także w transmisji bloków danych z 1 pamięci do 2, a także między urządzeniami I/O. Może być realizowany na 2 sposoby:

- burst mode – wyłączenie procesora na pełny czas transmisji danych
- cycle steal mode – wykradanie niektórych taktów zegarowych lub całych cykli maszynowych procesora w trakcie realizacji programu przez procesor, zwan również przeplataniem cykli procesora i transmisji DMA.

32. Jak procesor widzi układy I/O?

Każdy układ I/O jest widziany przez procesor jako zbiór rejestrów, z których każdy umieszczony jest pod określonym adresem w przestrzeni adresowej przydzielonej temu układowi.

33. Ile komórek pamięci ma procesor 8-bitowy?

Co to za zjebane pytanie jest

34. Co to znaczy, że układ jest programowalny?

Oznacza to, że do zapisania danych w pamięci programowalnej niezbędny jest wyspecjalizowany układ, nazywany programatorem.