

Pytania Testowe - Mikrokontrolery ATmega16

1. **Pytanie:** `LDI r16, 0b00000000 OUT DDRA, r16`
 - 0: wysyła 0b00000000 na port A.
 - **1: ustawia port A jako wejście. @@**
 - 2: ustawia port A jako wyjście.
 - 3: załącza rezystory pull-up portu A.
2. **Pytanie:** `LDI r16, 0b11111111 OUT DDRB, r16`
 - 0: wysyła 0b11111111 na port B.
 - 1: odłącza rezystory pull-up portu B.
 - 2: ustawia port B jako wejście.
 - **3: ustawia port B jako wyjście. @@**
3. **Pytanie:** `IN r16, PINA`
 - 0: ustawia stan wysokiej impedancji na liniach portu A.
 - 1: odczytuje kierunek (status) linii portu A (wejście czy wyjście).
 - 2: ustawia linie portu A jako wejścia.
 - **3: odczytuje stan linii portu A. @@**
4. **Pytanie:** W mikrokontrolerze ATmega16, wykorzystując transmisję UART kontrola parzystości może zostać wykonana:
 - **0: sprzętowo i/lub programowo. @@**
 - 1: do kontroli parzystości konieczne jest stosowanie bibliotek języka C.
 - 2: tylko programowo.
 - 3: nie ma możliwości kontroli parzystości.
5. **Pytanie:** Wektor (adres) przerwania od odebranego znaku przez UART, to:
 - 0: 0x0024
 - 1: 0x001A
 - 2: 0x0018
 - **3: 0x0016 @@**
6. **Pytanie:** W mikrokontrolerze ATmega16 prędkość transmisji szeregowej UART ustawiamy w rejestrach:
 - 0: UCSRB i UCSRC
 - 1: SPH i SPL
 - 2: UCSRA i UCSRB
 - **3: UBRRH i UBRRL @@**
7. **Pytanie:** Instrukcja `SBIS`:
 - **0: testuje stan wybranej linii portu I/O. @@**
 - 1: testuje status (kierunek) wybranej linii portu I/O.
 - 2: ustawia 1 na wybranej linii portu I/O.
 - 3: ustawia 0 na wybranej linii portu I/O.
8. **Pytanie:** Instrukcja `CPI`:
 - 0: porównuje zawartość dwóch rejestrów roboczych.
 - **1: porównuje zawartość rejestru roboczego ze stałą. @@**
 - 2: ustawia wszystkie linie I/O wybranego portu na 0.
 - 3: zmienia na przeciwny stan linie I/O wybranego portu.

9. **Pytanie:** Zakończenie wysyłania znaku (opróżnienie bufora) przez UART sygnalizowane jest ustawieniem bitu:
- 0: TXEN
 - 1: UCSZ0
 - **2: UDRE @@**
 - 3: RXC
10. **Pytanie:** Instrukcja **CBI**:
- 0: zmienia kierunek pracy wybranej linii portu.
 - **1: ustawia „0” na wybranej linii portu. @@**
 - 2: ustawia stan wysokiej impedancji na wybranej linii portu.
 - 3: neguje stan na wybranej linii portu.
11. **Pytanie:** Zakończenie odbierania znaku przez UART sygnalizowane jest ustawieniem bitu:
- **0: RXC @@**
 - 1: TXC
 - 2: TXEN
 - 3: RXEN
12. **Pytanie:** Maska bitowa nakładana jest poprzez zastosowanie instrukcji:
- 0: ROL
 - **1: AND @@**
 - 2: SUB
 - 3: ADD
13. **Pytanie:** Do globalnego zablokowania przerwań służy instrukcja:
- 0: SBI
 - **1: CEI (tu jest błąd poprawna to CLI) @@**
 - 2: SEI
 - 3: CPI
14. **Pytanie:** Jeżeli w rejestrze wpisana jest wartość 7 i zastosujemy maskę bitową 0000 0011 to otrzymamy:
- 0: 252
 - 1: 2
 - 2: 7
 - **3: 3 @@**
15. **Pytanie:** Pole danych w ramce UART może zawierać:
- **0: 5,6,7,8,9 bitów @@**
 - 1: 8,9 bitów
 - 2: 5,8,9 bitów
 - 3: 7,8 bitów
16. **Pytanie:** Producentem mikrokontrolerów AVR jest:
- 0: Intel
 - **1: Microchip @@**
 - 2: Motorola
 - 3: Philips
17. **Pytanie:** Instrukcja **ANDI r16, 0b00001111**
- 0: ustawia „1” na czterech młodszych bitach rejestru r16
 - 1: wpisuje 15 do rejestru r16
 - 2: dodaje 15 do liczby zapisanej w rejestrze r16

- **3: nakłada maskę bitową na zawartość rejestru r16 @@**
18. **Pytanie:** Do powrotu z obsługi przerwania można zastosować sekwencję instrukcji:
- **0: RET oraz BSET(7) @@**
 - 1: RET oraz SET
 - 2: RET oraz CLZ
 - 3: RET oraz CLI
19. **Pytanie:** Wektor (adres) przerwania od nadanego znaku przez UART, to:
- 0: 0x0016
 - **1: 0x001A @@**
 - 2: 0x0020
 - 3: 0x0014
20. **Pytanie:** Instrukcja **RJMP**:
- 0: zastępuje przestarzałą instrukcję JMP w nowszych procesorach.
 - **1: wykonuje skok do adresu w +/- 2k. @@**
 - 2: nie może być stosowana w procesorach posiadających więcej niż 8 linii I/O.
 - 3: może być stosowana wymiennie z instrukcją JMP do adresowania pamięci powyżej 8k.
21. **Pytanie:** Maksymalna częstotliwość taktowania mikrokontrolera ATmega16 to:
- 0: 4 MHz
 - 1: 32 MHz
 - 2: 20 MHz
 - **3: 16 MHz @@**
22. **Pytanie:** Napięcia zasilania układu ATmega16L to:
- 0: 3,3 V - 5,0 V
 - 1: 3 V - 15 V
 - **2: 2,7 V - 5,5 V @@**
 - 3: 4,0 V - 5,0 V
23. **Pytanie:** Producent układu ATmega16 zapewnia liczbę cykli zapisu/kasowania pamięci programu:
- 0: 100 000
 - 1: 1 000 000
 - 2: 1 000
 - **3: 10 000 @@**
24. **Pytanie:** Producent układu ATmega16 zapewnia liczbę cykli zapisu/kasowania pamięci EEPROM:
- 0: 100
 - 1: 10 000
 - 2: 1 000
 - **3: 100 000 @@**
25. **Pytanie:** Mikrokontroler ATmega16 ma:
- 0: osiem 1-kanalowych przetworników a/c
 - 1: osiem 1-kanalowych, dwa 2-kanalowe przetworniki a/c
 - **2: jeden 8-kanalowy przetwornik a/c @@**
 - 3: osiem 7-kanalowych przetworników a/c
26. **Pytanie:** Wynik przetwarzania A/C możemy odczytać w pamięci (danych) pod adresem:

- 0: 0x005D i 0x005E
 - 1: 0x0060 i 0x0061
 - **2: 0x0024 i 0x0025 @@**
 - 3: wynik nie jest umieszczany w pamięci danych
27. **Pytanie:** Czas konwersji przetwornika A/C dla ATmega16, to:
- 0: 1–2 s
 - **1: 13–260 μ s @@**
 - 2: 2–13 ms
 - 3: 160–300 ms
28. **Pytanie:** Wystąpienie przerwania od odebranego znaku wskazywane jest przez znacznik (bit):
- 0: RXCIE
 - 1: RXEN
 - 2: RXB8
 - **3: RXC @@**
29. **Pytanie:** Układy buforowe (bramki buforowe) stosuje się na wyjściach mikrokontrolera, m.in. w celu:
- 0: dodatkowego uziemienia sterowanych podzespołów.
 - 1: przyspieszenia pracy układów peryferyjnych.
 - 2: uniezależnienia stanu wyjścia od zmian temperatury.
 - **3: zwiększenia wydajności prądowej wyjścia. @@**
30. **Pytanie:** Wewnętrzne napięcie referencyjne w ATmega16 ma wartość:
- 0: 1,28 V
 - 1: 3,33 V
 - **2: 2,56 V @@**
 - 3: 5,12 V
31. **Pytanie:** Na wybranych wejściach przetwornika A/C można wstępnie wzmocnić sygnał:
- 0: 10x, 20x
 - 1: 2x, 10x
 - **2: 10x, 200x @@**
 - 3: 4x, 10x
32. **Pytanie:** Aby rozpocząć konwersję analogowo-cyfrową ustawiamy bit:
- 0: ADIF
 - 1: ADLAR
 - **2: ADSC @@**
 - 3: ADEN
33. **Pytanie:** Do wyboru źródła referencyjnego w przetworniku A/C służą bity:
- 0: ADPS2:0
 - 1: MUX4:0
 - **2: REFS1:0 @@**
 - 3: ADC9:0
34. **Pytanie:** W ATmega16 (f-częstotliwość taktowania mikrokontrolera) największa możliwa do ustawienia częstotliwość sygnału zegarowego SPI to:
- 0: f/4
 - **1: f/2 @@**
 - 2: f/8
 - 3: f/16

35. **Pytanie:** Przetwornik A/C w ATmega16 to przetwornik:
- 0: 16-bitowy.
 - 1: 14-bitowy.
 - 2: 8-bitowy.
 - **3: 10-bitowy. @@**
36. **Pytanie:** Zakończenie transmisji znaku przez SPI sygnalizowane jest ustawieniem znacznika (bitu):
- 0: SPOL
 - **1: SPIF @@**
 - 2: SPE
 - 3: DORD
37. **Pytanie:** W układzie MASTER linie interfejsu SPI ustawione są:
- 0: MOSI - wy MISO - wy SCK - wy SS- wy
 - 1: MOSI - wy MISO - w y SCK - wy SS-we
 - 2: MOSI - wy MISO - we SCK - we SS - w
 - **3: MOSI - wy MISO - we SCK- wy SS- wy @@**
38. **Pytanie:** Podczas transmisji SPI z układu MASTER do wielu układów SLAVE sygnał zegarowy może być generowany przez:
- **0: tylko układ MASTER w którym ustawiona znacznik (bit) MSTR=1 @@**
 - 1: układ MASTER lub aktywny układ SLAVE w zależności od konfiguracji znacznika (bitu) DORD
 - 2: wybrany układ SLAVE aktywowany niskim poziomem sygnału SS
 - 3: tylko ukad SLAVE w którym wstępnie skonfigurowano znacznik (bit) MSTR=0
39. **Pytanie:** Typowa dopuszczalna wartość prądu linii portu I/O mikrokontrolera ATmega16 to:
- 0: 200 mA
 - 1: 2 A
 - **2: 20 mA @@**
 - 3: 2 mA
40. **Pytanie:** Linie interfejsu SPI to:
- **0: PB7, PB6, PB5, PB4 @@**
 - 1: PD3, PD2, PD1, PD0
 - 2: PA5, PA4, PA3, PA2
 - 3: definiowane przez użytkownika linie portu PD lub PB
41. **Pytanie:** Jaki jest efekt działania instrukcji `LDI r16, 10`?
- 0: Zwiększa zawartość komórki danych pod adresem \$16 o wartość 10
 - **1: Zapisuje wartość 10 do rejestru r16 @@**
 - 2: Zamienia liczbę zapisaną w systemie szesnastkowym na liczbę w systemie dziesiętnym
 - 3: Zapisuje wartość 10 do kolejnych 16 komórek pamięci danych
42. **Pytanie:** Jaki jest efekt działania dyrektywy `.include`?
- **0: Dołącza do projektu zewnętrzny plik tekstowy @@**
 - 1: Automatycznie uruchamia kompilację programu
 - 2: Umożliwia przesłanie kodu programu do symulatora
 - 3: Zezwala na kompilację kodu assemblera na język C
43. **Pytanie:** Jaki jest efekt działania dyrektywy `.cseg`?

- 0: Włączenie automatycznego uzupełniania instrukcji assemblerowych
 - 1: Umożliwienie automatycznej kompilacji kodu programu assemblerowego
 - **2: Wybór pamięci programu do umieszczenia kodu programu @@**
 - 3: Podział (segmentacja) obszaru pamięci danych na bloki
44. **Pytanie:** Gdzie zapisywany jest kod programu?
- 0: W pamięci zewnętrznej nieulotnej i po resecie jest przepisywany do pamięci danych
 - **1: W pamięci programu @@**
 - 2: W pamięci danych
 - 3: W pamięci EEPROM i po resecie jest przepisywany do pamięci danych
45. **Pytanie:** Jaki jest rozmiar pamięci programu w mikrokontrolerze ATmega16?
- 0: 32MB (2 bloki pamięci po 16MB)
 - 1: Zależy od stopnia wykorzystania pamięci danych, max. 16*8kB
 - 2: 1MB pamięci wewnętrznej i 6MB pamięci zewnętrznej
 - **3: 16kB pamięci non-volatile @@**
46. **Pytanie:** Ile taktów zegarowych trwa instrukcja **LDI**?
- **0: 1 takt zegarowy @@**
 - 1: 3 takty zegarowe
 - 2: 3 takty - gdy wykorzystujemy oscylator wbudowany, 12 taktów - gdy wykorzystujemy rezonator (kwarce) zewnętrzny
 - 3: 12 taktów zegarowych
47. **Pytanie:** Liczba 164 (dziesiętnie) to w zapisie szesnastkowym (hexadecymalnym):
- **0: A4 @@**
 - 1: B8
 - 2: 92
 - 3: 4F
48. **Pytanie:** Gdzie zostanie umieszczony wynik działania instrukcji **ADD r16, r17**?
- 0: W pamięci danych pod adresem wskazanym przez RAMEND
 - 1: W kolejnym rejestrze r18
 - **2: W rejestrze r16, wcześniejsza wartość rejestru zostaje nadpisana @@**
 - 3: W pierwszej wolnej komórce pamięci danych
49. **Pytanie:** Do czego służy bit (znacznik) C w rejestrze SREG?
- **0: Informuje o przekroczeniu zakresu liczb 8-bitowych @@**
 - 1: Znacznik jest ustawiany, gdy do rejestru wpisujemy tę samą wartość
 - 2: Ustawienie znacznika pozwala dołączyć do kompilacji wstawki języka C
 - 3: Znacznik jest ustawiany, gdy próbujemy zapisać dane do niepełnego rejestru
50. **Pytanie:** Do czego służy instrukcja **LDI**?
- **0: Pozwala zapisać liczbę z zakresu -255 do jednego z rejestrów r16-r31 @@**
 - 1: Przepisuje zawartość jednego z rejestrów r0-r31 do innego rejestru
 - 2: Wpisuje liczbę 16-bitową do jednego z rejestrów r0-r31
 - 3: Dokonuje konwersji liczby 16-bitowej na 8-bitową i zapisuje w jednym z rejestrów r0-r31
51. **Pytanie:** Liczba 183 (dziesiętnie) to w zapisie binarnym assemblera:
- **0: 0b10110111 @@**
 - 1: 0b10010110

- 2: 0b01111011
 - 3: 0b01110111
52. **Pytanie:** Która deklaracja etykiety jest niepoprawna:
- **0: 10ETYKIETA @@**
 - 1: ETYKIETA10
 - 2: ETYKIETA_10
 - 3: Etykieta10
53. **Pytanie:** Gdzie umieszczone są rejestry r0–r31 (robocze)?
- 0: Programista decyduje w której pamięci je umieścić
 - **1: W pamięci danych (SRAM) @@**
 - 2: W pamięci programu
 - 3: Młodsze r0–r15 w pamięci danych, starsze r16–r31 w pamięci programu
54. **Pytanie:** Jaką rolę w strukturze procesora pełni Program Counter (licznik rozkazów)?
- 0: Zlicza liczbę rozkazów w programie
 - **1: Wskazuje adres następnej instrukcji w pamięci programu @@**
 - 2: Zlicza liczbę wykonanych instrukcji od rozpoczęcia programu
 - 3: Wskazuje liczbę taktów zegara od ostatniego resetu procesora
55. **Pytanie:** Do czego służy bit (znacznik) Z w rejestrze SREG?
- 0: Ustawienie znacznika pozwala programowo zresetować procesor
 - 1: Znacznik jest ustawiany, jeżeli rejestr do którego próbujemy wpisać wartość jest zabezpieczony przed zapisem
 - 2: Ustawienie znacznika anuluje działanie ostatnio wykonanej operacji arytmetyczno-logicznej
 - **3: Informuje o uzyskaniu wartości zerowej po wykonaniu ostatniej operacji arytmetyczno-logicznej @@**
56. **Pytanie:** Jaki jest efekt działania instrukcji **INC r16**?
- **0: Zwiększa zawartość rejestru r16 o 1 @@**
 - 1: Zeruje zawartość rejestru r16
 - 2: Przepisuje do rejestru r16 aktualną zawartość rejestru SREG
 - 3: Sprawdza dostępność do zapisu komórki pamięci danych o adresie \$16
57. **Pytanie:** W jakim zakresie obszaru pamięci programu jest możliwe wykonanie skoku bezwarunkowego z wykorzystaniem instrukcji JMP (dla ATmega16)?
- 0: 16MB
 - **1: Cała dostępna pamięć programu @@**
 - 2: Przestrzeń zależna jest od wartości RAMEND
 - 3: 216 bajtów
58. **Pytanie:** Pamięć programu mikrokontrolera ATmega16 zorganizowana jest w słowa:
- 0: 32-bitowe
 - 1: 8-bitowe
 - 2: 8-, 16- lub 32-bitowe, w zależności od długości zastosowanej instrukcji
 - **3: 16-bitowe @@**
59. **Pytanie:** W instrukcjach przesunięcia (np. ROL, LSL, itp.) znacznik C wykorzystywany jest do:
- 0: Zmiany kierunku przesuwania
 - 1: Przywrócenia pierwotnej wartości rejestru
 - **2: Przechowywania „wysuniętego” bitu @@**
 - 3: Informowania o zakończeniu przesunięcia całego bajtu

60. **Pytanie:** Wynik działania instrukcji **MUL r16, r17** umieszczony jest:
- **0: w rejestrach r0 i r1 @@**
 - 1: w rejestrach r16 i r17
 - 2: młodszy bajt – w r16, starszy bajt – decyduje programista
 - 3: w rejestrach r18 i r19
61. **Pytanie:** Do czego służy bit (znacznik) N w rejestrze SREG?
- 0: Znacznik jest ustawiany, gdy wynik obliczeń jest liczbą naturalną
 - 1: Znacznik jest ustawiany, gdy próbujemy zapisać dane do rejestru tylko do odczytu
 - 2: Ustawienie znacznika neguje wynik obliczeń
 - **3: Znacznik informuje o uzyskaniu liczby ujemnej @@**
62. **Pytanie:** Do czego służy bit (znacznik) T w rejestrze SREG?
- 0: Ustawienie znacznika uruchamia wewnętrzny układ czasowy (timer)
 - 1: Znacznik jest ustawiany, gdy występuje przepełnienie pamięci danych
 - 2: Znacznik jest ustawiany, gdy program pracuje w pętli
 - **3: Jest to uniwersalny akumulator 1-bitowy @@**
63. **Pytanie:** Do czego służy instrukcja **CPSE**?
- 0: Porównuje dwa rejestry, skacze pod wskazany adres gdy warunek jest spełniony
 - 1: Porównuje rejestr r0–r31 z rejestrem SREG, wykonuje skok gdy warunek jest spełniony
 - 2: Porównuje rejestr i stałą, wykonuje skok zależnie od spełnienia warunku
 - **3: Porównuje dwa rejestry, pomija kolejną instrukcję gdy warunek jest spełniony @@**
64. **Pytanie:** Czy instrukcja **BREQ** uwzględnia stan znacznika C?
- 0: Tak – pod warunkiem, że ustawiono wcześniej znacznik T
 - 1: Tak
 - **2: Nie @@**
 - 3: Tak, jeżeli przekroczono zakres liczb 8-bitowych
65. **Pytanie:** Ile taktów zegarowych trwa instrukcja **BRNE**?
- 0: 4 takty
 - **1: 1 lub 2 takty @@**
 - 2: 3 takty zegarowe
 - 3: 12 taktów
66. **Pytanie:** Rejestr SREG jest rejestrem?
- **0: 8-bitowym @@**
 - 1: wielkość jest ustalana programowo od 4 do 16 bitów
 - 2: 7-bitowym
 - 3: 8 lub 16-bitowym, w zależności od ustawienia flagi H
67. **Pytanie:** Rejestr SREG służy do:
- 0: zwiększenia pojemności ALU i wykonywania działań na liczbach 16-bitowych
 - 1: tymczasowego zapisania wyniku obliczeń ALU
 - **2: wskazania stanu po ostatnio wykonanej instrukcji przez ALU @@**
 - 3: przechowywania kopii ostatnio wykonanej przez ALU instrukcji
68. **Pytanie:** Jednostka ALU to:

- 0: wydzielony blok w jednostce centralnej do zapisywania wyników obliczeń arytmetyczno-logicznych
 - **1: podzespół procesora służący do wykonywania operacji arytmetycznych i logicznych @@**
 - 2: blok pamięci danych wydzielony do obliczeń arytmetyczno-logicznych
 - 3: zaimplementowany w CPU podprogram służący do obliczeń arytmetyczno-logicznych
69. **Pytanie:** W procesorze 8-bitowym, programowe opóźnienie dłuższe niż 216 ms:
- 0: nie jest możliwe programowo, trzeba stosować sprzętowe liczniki (timery)
 - 1: jest możliwe, ale wymaga programowania w języku C
 - **2: jest możliwe do realizacji @@**
 - 3: nie jest możliwe, wymagana jest jednostka ALU 16-bitowa
70. **Pytanie:** Gdzie znajduje się rejestr SREG?
- **0: w pamięci danych, w rejestrach specjalnych @@**
 - 1: w pamięci danych, w rejestrach wskaźnikowych
 - 2: w pamięci programu, pod adresem 0x0030
 - 3: w pamięci programu, w obszarze wektorów przerwań
71. **Pytanie:** W której pamięci lokowany jest stos?
- 0: w pamięci programu
 - 1: w pamięci danych EEPROM
 - **2: w pamięci danych SRAM @@**
 - 3: programista decyduje w której pamięci
72. **Pytanie:** Stos lokowany jest:
- 0: na początku wolnej przestrzeni pamięci
 - 1: zawsze na początku pamięci
 - 2: zawsze na końcu pamięci ??
 - **3: programista decyduje gdzie lokowany jest stos @@**
73. **Pytanie:** Instrukcja **LDI r16, low(0x045F)**:
- 0: zapisuje do r16 zawartość komórki wskazanej przez 0x045F
 - 1: zapisuje do r16 zawartość komórki 0x045F
 - **2: ładuje do r16 wartość 0x5F @@**
 - 3: zapisuje zawartość r16 pod adres 0x045F
74. **Pytanie:** Wskaźnik stosu:
- **0: wskazuje adres komórki w pamięci danych do której zapisano adres powrotu z podprogramu . @@**
 - 1: wskazuje wielkość (liczbę komórek) przeznaczoną na stos.
 - 2: wskazuje wielkość (liczbę komórek) stosu wykorzystaną w przez podprogram.
 - 3: wskazuje adres komórki w pamięci danych do zapisu adresu powrotu z podprogramu.
75. **Pytanie:** Przy zapisie na stos:
- 0: programista decyduje jak zachowa się SP.
 - 1: wskaźnik stosu SP jest zwiększany.
 - **2: wskaźnik stosu SP jest zmniejszany. @@**
 - 3: wskaźnik stosu SP pozostaje bez zmian.
76. **Pytanie:** Instrukcja **CALL**:
- **0: zmniejsza wskaźnik stosu. @@**

- 1: nie wpływa na wartość wskaźnika stosu.
- 2: zwiększa wskaźnik stosu.
- 3: może zmniejszać lub zwiększać stan wskaźnika stosu.

77. **Pytanie:** Instrukcja **RET** powoduje:

- 0: powrót do adresu zapisanego w SP
- 1: powrót do początku podprogramu.
- 2: skok do początku programu.
- **3: skok do adresu ostatnio zapisanego na stosie. @@**

78. **Pytanie:** W procesorze 8-bitowym wykonanie zagnieżdżonych podprogramów:

- **0: jest możliwe, ale zależy od wielkości przestrzeni przewidzianej na stos. @@**
- 1: nie jest możliwe, za mała przestrzeń adresowa.
- 2: jest możliwe tylko 2-krotne zagnieżdżenie.
- 3: nie jest możliwe stosując tylko instrukcje asemblera.

79. **Pytanie:** W procesorze ATmega16 rozmiar rejestru SP (Stack Pointer) to:

- 0: 9 bitów
- 1: 11 bitów
- **2: 16 bitów @@**
- 3: 128 bitów

80. **Pytanie:** W ATmega16 jednoczesne stosowanie podprogramów i makr:

- **0: jest możliwe. @@**
- 1: jest możliwe, ale powoduje konflikt adresów w pamięci danych.
- 2: nie jest możliwe przy programowaniu w asemblerze.
- 3: nie jest możliwe w procesorze 8-bitowym.

81. **Pytanie:** W mikrokontrolerze ATmega16 wskaźnik stosu (SP) umieszczony jest:

- 0: na początku wolnej przestrzeni pamięci
- 1: zawsze na końcu pamięci programu
- **2: w obszarze rejestrów specjalnych pod adresem 0x3D i 0x3E @@**
- 3: programista decyduje pod jakim adresem umieścić wskaźnik stosu (SP)

82. **Pytanie:** Rejestr wskaźnikowy X tworzą rejestry:

- **0: r26 i r27. @@**
- 1: r16 i r17.
- 2: r30 i r31.
- 3: r0 i r1.

83. **Pytanie:** Jaką wartość przyjmie licznik rozkazów po wykonaniu ostatniej instrukcji

(RET) ? **push r16 ret**

- **0: 0x0201 @@**
- 1: 0x0200
- 2: 0x045F
- 3: 0x0001

84. **Pytanie:** Obszar tzw. wektorów przerwań to:

- **0: obszar pamięci programu w którym umieszczone są adresy obsługi przerwań @@**
- 1: obszar pamięci danych przeznaczony do zapisu danych po wystąpieniu przerwania
- 2: obszar pamięci danych zawierający kopie rejestrów roboczych, na wypadek przerwania pracy mikrokontrolera

- 3: obszar pamięci danych, w którym zapisywane są ustawienia mikrokontrolera niezbędne do wznowienia pracy po nieoczekiwanym przerwaniu programu

85. **Pytanie:** Rejestry I/O umieszczone są w:

- **0: pamięci danych SRAM @@**
- 1: pamięć programu
- 2: w obszarze wektorów przerwań
- 3: pamięć danych EEPROM

Pytanie 1 Podczas transmisji SPI z układu MASTER do wielu układów SLAVE sygnał zegarowy może być generowany przez:

-
- 2 – zapytania zostały przetworzone

Oto pytania i poprawne odpowiedzi z przesłanych plików PDF, przedstawione w wymaganym formacie.

Pytania i Odpowiedzi z Plików PDF

Pytanie 1 Podczas transmisji SPI z układu MASTER do wielu układów SLAVE sygnał zegarowy może być generowany przez:

- **Poprawna odpowiedź:** tylko układ MASTER w którym ustawiona znacznik (bit) MSTR=1
-

Pytanie 2 Która deklaracja etykiety jest niepoprawna:

- **Poprawna odpowiedź:** 10 ETYKIETA (Etykieta nie może zaczynać się cyfrą)
-

Pytanie 3 Rejestr SREG służy do:

- **Poprawna odpowiedź:** wskazania stanu po ostatnio wykonanej instrukcji przez ALU
-

Pytanie 4 Przy zapisie na stos:

- **Poprawna odpowiedź:** wskaźnik stosu SP jest zmniejszany
 -
-

Pytanie 5 Jaki jest efekt działania dyrektywy `.cseg`?

- **Poprawna odpowiedź:** wybór pamięci programu do umieszczenia kodu programu
 -
-

Pytanie 6 Rejestry robocze umieszczone są:

- **Poprawna odpowiedź:** w pamięci danych SRAM
 -
-

Pytanie 7 Wskaźnik stosu:

- **Poprawna odpowiedź:** wskazuje adres komórki w pamięci danych do zapisu adresu powrotu z podprogramu.
 -
-

Pytanie 8 Instrukcja `LDI r16, low (0x045F)`:

- **Poprawna odpowiedź:** ładuje do r16 wartość 0x5F
 -
-

Pytanie 9 Podaj (w postaci dziesiętnej) wartość obliczoną dla przyspieszonej transmisji asynchronicznej (Double Speed Mode), dla prędkości 2400 baud i częstotliwości zegara $f_{OSC}=7\text{ MHz}$.

- **Wzór:** $UBRR=8BAUDf_{OSC}-1$
 -
-

Pytanie 10 Obszar tzw. wektorów przerwań to:

- **Poprawna odpowiedź:** obszar pamięci programu w którym umieszczone są adresy obsługi przerwań
 -
-

Pytanie 11 Gdzie zostanie umieszczony wynik działania instrukcji `ADD r16, r17`?

- **Poprawna odpowiedź:** w rejestrze r16, wcześniejsza wartość rejestru zostaje nadpisana
 -
-

Pytanie 12 Wykonanie instrukcji: `Idi r16, 0b00000000 out DDRA, r16`

- **Poprawna odpowiedź:** ustawia port A jako wejście
 -
-

Pytanie 13 Wynik działania instrukcji `MUL r16, r17` umieszczony jest:

- **Poprawna odpowiedź:** w rejestrach r1 i r0
 -
-

Pytanie 14 Jaką rolę w strukturze procesora pełni Program Counter (licznik rozkazów)?

- **Poprawna odpowiedź:** wskazuje adres następnej instrukcji w pamięci programu
 -
-

Pytanie 15 W mikrokontrolerze ATmega16 wskaźnik stosu (SP) umieszczony jest:

- **Poprawna odpowiedź:** w obszarze rejestrów specjalnych pod adresem 0x3D i 0x3E
 -
-

Pytanie 16 Rejestr wskaźnikowy X tworzą rejestry:

- **Poprawna odpowiedź:** r26 i r27.
 -
-

Pytanie 17 W mikrokontrolerze ATmega16 prędkość transmisji szeregowej UART ustawiamy w rejestrach:

- **Poprawna odpowiedź:** UBRRH i UBRRL
 -
-

Pytanie 18 Wykonanie instrukcja **CALL**:

- **Poprawna odpowiedź:** zmniejsza wskaźnik stosu
 -
-

Pytanie 19 W instrukcjach przesunięcia (np. ROL, LSL, itp.) znacznik C wykorzystywany jest do:

- **Poprawna odpowiedź:** przechowywania "wysuniętego" bitu
 -
-

Pytanie 20 Zakończenie wysyłania znaku (opróżnienie bufora) przez UART sygnalizowane jest ustawieniem bitu:

- **Poprawna odpowiedź:** UDRE
 -
-

Pytanie 21 Do powrotu z obsługi przerwania można zastosować sekwencje instrukcji:

- **Poprawna odpowiedź:** RET oraz BSET(7)
 -
-

Pytanie 22 W programie głównym MAIN odbieramy i nadajemy znaki wywołując odpowiednie podprogramy USART_Receive i USART_Transmit przekazując dane poprzez rejestr r16. Odbieramy wartość 1 (dziesiętnie). Jaka wartość zostanie odesłana (wpisz dziesiętnie)? *Kod:*

Fragment kodu

MAIN:

ldi r17, 0x0a

call USART_Receive

add r16, r17

call USART_Transmit

jmp MAIN

- **Poprawna odpowiedź:** 11 (*Obliczenie: $r17=0x0A=10$. Po odbiorze $r16=1$. Po dodawaniu $r16=1+10=11$*)
 -
-

Pytanie 23 Instrukcja **RET** powoduje:

- **Poprawna odpowiedź:** skok do adresu ostatnio zapisanego na stosie
 -
-

Pytanie 24 Podaj (w postaci dziesiętnej) wartość obliczoną dla przyspieszonej transmisji asynchronicznej (Double Speed Mode), dla prędkości 2400 baud i częstotliwości zegara $f_{OSC}=6$ MHz.

- **Wzór:** $UBRR=8BAUDf_{OSC}-1$
 -
-

Pytanie 25 Jaki jest czas wysłania całej ramki UART (wysyłamy 1 bajt) dla ustawionych parametrów transmisji 1200,8,N,1 i częstotliwości taktowania zegara $f_{osc}=8$ MHz? Wynik podaj w mili sekundach.

- **Poprawna odpowiedź:** 8 (*Uzasadnienie: ramka ma 1 (start)+8 (dane)+1 (stop)=10 bitów. Czas transmisji: $1200 \text{ Baud} \cdot 10 \approx 8,33 \text{ ms}$*)
 -
-

Pytanie 26 W trybie MASTER SPI, kierunki sygnałów na pinach portu to:

- **Poprawna odpowiedź:** MOSI - wy, MISO-we, SCK-wy, SS-wy
 -
-

Pytanie 27 W ATmega16 jednoczesne stosowanie podprogramów i makr:

- **Poprawna odpowiedź:** jest możliwe
 -
-

Pytanie 28 Jakią wartość przyjmie licznik rozkazów po wykonaniu ostatniej instrukcji (**RET**)?

- **Ustawienia początkowe:** $SP = 0x0102$ (z instrukcji **out SPH, r29; out SPL, r28** i założeniu $r29=0x01, r28=0x02$)
-

Obliczenie: $UBRR=8 \times 2400 / (6\,000\,000 - 1) \approx 363.58$

Poprawna odpowiedź: 364

Obliczenie: $UBRR=8 \times 2400 / (6\,000\,000 - 1) \approx 311.5$

Poprawna odpowiedź: 312

Poprawna odpowiedź: 0x0201 (*Adres 0x0201 jest adresem instrukcji następującej po `call SUB1`, który zostaje ściągnięty ze stosu przez `RET`*)