

Temat:

Minimalny system operacyjny (BIOS).

1. Zagadnienia podstawowe.

Sercem układu STRC52 jest mikrokontroler AT89C52 (zgodny z 80C52). Posiada on pamięć Flash EPROM. Pamięć ta zawiera program monitora. W dotychczasowej wersji działanie tego programu ogranicza się do prostych procedur wspomagających uruchamianie tworzonego w ramach zajęć oprogramowania. Zadaniem stawianym w ramach tych ćwiczeń jest zbudowanie zestawu funkcji (API) wspomagających pracę systemu i dających wsparcie przyszłym użytkownikom-programistom.

2. Procedury testowania sprzętu i określania konfiguracji.

Po włączeniu zasilania lub po chwilowym wprowadzeniu mikrokontrolera w stan RESET, dobrze zaprojektowany system powinien wykonać szereg czynności, procedury te nazywane są POST (od ang. Power On Self Test). Obecnie wbudowany program monitora ma te funkcje dla uproszczenia działania systemu wyłączone.

Do najważniejszych czynności związanych z procedurami POST dla systemu STRC52 można zaliczyć:

a) testowanie pamięci kodu monitora (weryfikacja sum kontrolnych):

- stałej (wewnętrzna pamięć EPROM - symulowana w zadaniu obszarem pamięci CODE),

- nie ulotnej (SRAM z podtrzymywaniem baterijnym i pamięci w urządzeniach zewnętrznych np.: RTC),

Uwaga! Kod monitora (tworzony na ćw.2), jest na potrzeby ćwiczenia umieszczony w pamięci SRAM, ale dostęp do niego jest z poziomu obszaru CODE.

b) testowanie obszarów pamięci ulotnej (sprawdzenie czy jest funkcjonalnie poprawna, czy dalsza współpraca będzie bez awaryjna):

- prosty zapis fragmentów pamięci, po czym jej weryfikacja (znane są np.: algorytmy kroczącej jedynki lub kroczącego zera) - taki test sprawdza z dużym prawdopodobieństwem, czy dany obszar pamięci działać będzie poprawnie,

- w przypadku STRC52, testowanie powinno dotyczyć obszarów w których jest pewność że nie rezyduje tam program monitora (dokładnie jego replika tworzona na ćw.2), funkcje POST, funkcje usługowe systemu ani żadne dane systemowe, czyli:

0x0000 - 0x3FFF - pamięć danych ogólnego przeznaczenia

0x4000 - koniec kodu - replika monitora tworzona na ćw.2 oraz procedury POST i usługowe tworzone na tym ćwiczeniu.

0x8000 - 0xFFFF - obszar niedostępny - używany przez urządzenia I/O

Więcej szczegółów zawarto w tabeli.1 w dalszej części dokumentu.

c) testowanie RTC (PCF8583):

- głównie sprawdzenie czy urządzenie jest obecne,

- czy przechowuje poprawne dane w polu czas i w reszcie pamięci (w której można przechowywać ustawienia systemu) - sugerowane jest aby ten obszar zabezpieczyć sumą kontrolną, podobnie jak wszystkie dane w obszarach nie ulotnych a modyfikowalnych,

Procedury liczenia sum kontrolnych są dobrze opisane w literaturze. Treść wybranych implementacji załączono na końcu dokumentu. W przypadku liczenia sum kontrolnych małych bloków (mniej niż 256B), lepiej i wydajniej jest używać algorytmów podobnych do CRC-8. Dla większych bloków algorytmy te nie są wiarygodne i sugerowane jest używanie algorytmów podobnych do CRC-16 (ze względu na większą odporność na pojawiające się błędy).

3. Procedury ładowania i archiwizacji danych i kodu.

System powinien umożliwiać zarówno załadowanie kodu jak i danych (tematykę tą omawiano szerzej w ćw. 2). Miejsce przeznaczenia nie powinno być ustalane arbitralnie, a przez odpowiednie parametry wywołania zaimplementowanych funkcji, np.:

```
char LoadHEX(char *s, unsigned int start, unsigned int len);
```

Gdzie:

*s - nazwa pliku na jaki czekano (nazwa ta identyfikuje dane po stronie PC, otrzymany plik HEX winien mieć taką nazwę)

start-wskaże gdzie procedura ładująca może umieszczać otrzymane przez RS232 dane z pliku HEX

len -w jak długim obszarze może umieszczać te dane

Funkcja powinna zwracać wynik swojej operacji, np.:

0 -sukcess

-1 -błąd przekroczenia obszaru podanego jako parametry (innymi słowy dane w pliku HEX wysłane musiały by trafić do innych obszarów niż dozwolone)

-2 -błąd sumy kontrolnej w odbieranych danych

-3 -nie otrzymano rekordu końca pliku HEX w określonym czasie (ang. TIMEOUT)

Pliki HEX zawiera informacje o miejscu przeznaczenia danych, ale procedura ładująca wywoływana przez system sama wie gdzie dane mogą trafić i jeżeli nie ma zgodności trzeba przerwać ładowanie.

Innym zagadnieniem wartym zaimplementowania jest możliwość kopiowania obszarów pamięci w STRC51 do komputera PC. Dzięki takiej funkcjonalności możliwe stałoby się zdalne (z poziomu PC) modyfikowanie pamięci (także konfiguracji) w STRC51. Przykładowo funkcja mogła by mieć prototyp:

```
char SaveHEX(char *s, unsigned int start unsigned int len);
```

Gdzie:

*s - nazwa pliku pod jaką system PC miałby zachować dane

start- miejsce w pamięci STRC51 skąd przenosić dane do PC

len - ile tych danych winno być skopiowanych

Należy pamiętać że w STRC51 obszar CODE i XDATA są identyczne.

4.Funkcje usługowe systemu operacyjnego

Zestaw funkcji zaimplementowanych w systemie, może być udostępniony programom użytkownika. Jest dobrą praktyką aby wykorzystywać o ile to możliwe funkcje systemowe a nie tworzyć je gdy jest taka możliwość, od podstaw w każdej aplikacji. Takie postępowanie wpływa znacząco na podniesienie bezpieczeństwa w systemie jak również mocno wpływa na zmniejszenie kodu wynikowego aplikacji uruchamianych w takim systemie.

Istotny jest sposób wywoływania takich funkcji. Najprostszą metodą jest ustalenie pewnego adresu w pamięci, pod którym znajdzie się tablica zawierająca wskaźniki do funkcji usługowych. Dla przykładu można by utworzyć tablicę:

```
xdata at 0x7f00 void (*xdata tab[MAX_SYSCALL])(void);
```

gdzie MAX_SYSCALL to liczba funkcji udostępnianych przez OS aplikacji użytkowej, oraz do takiej tablicy podstawić adresy funkcji systemowych. Deklaracja miejsca składowania takiej struktury jest nie przypadkowa i dla STRC52 jest to koniec obszaru pamięci SRAM, gdzie jest największe w ten sposób zostanie nie naruszona. Dla poprawnej kompilacji i działania, kod systemowy pod adresem 0x7f00 musi odpowiednio wstawić adresy udostępnianych procedur systemowych, natomiast aplikacja użytkowa nie może tego obszaru zniszczyć nadpisując go własnymi danymi.

W aplikacji użytkowej wywołanie takich funkcji mogło by wyglądać następująco:

```
...  
(*tab[OS_SYSCALL1])();
```

```
...
```

Gdzie OS_SYSCALL1 jest identyfikatorem wywoływanej funkcji.

W takim rozwiązaniu największy problem jest z przekazywaniem argumentów i zwracaniem wyników. Funkcje mogą być pisane i używane w różnych językach, i kompilowane różnymi narzędziami. W takich sytuacjach trudno jest definiować ogólny sposób komunikacji z funkcjami systemowymi. Najprostszym rozwiązaniem jest użycie wspólnego obszaru pamięci gdzie informacje będą umieszczane zapewniając komunikację między kodem "wołającym" a "wołanym" (czyli systemowym) Wtedy to funkcje systemowe będą mogły mieć prototyp:

Zatem proponuje się zadeklarowanie (podobnie jak zmiennej tab), zmiennej buffer:

```
xdata at 0x7e00 unsigned char buffer[MAX_BUFFER];
```

Odtąd kod aplikacji użytkowej może wpierv wpisać coś do takiej struktury a potem wywołać funkcję systemową:

```

...
strcpy(user_buffer, s);
(*tab[OS_SYSCALL6])();
...

```

Rozwiązanie takie choć bardzo proste, nie daje pełnej uniwersalności przekazywania argumentów jaką daje język "C". Problem ten można rozwiązać za pomocą mieszania języka "C" z assemblerowymi fragmentami kodu.

Warto zwrócić uwagę że dla powyższych przykładów obszar od 0x7e00 do 0x7fff winien być zarezerwowany i umiejętnie wykorzystywany w funkcjach aplikacji użytkowej. W systemach wyposażonych w układy MMU kontrolę nad takimi fragmentami sprawuje sprzęt, nie dopuszczając np.: do zamazania tablicy adresów funkcji systemowych.

4.1. Interfejs użytkownika – wejście / wyjście

Tematyka poruszana była w ramach poprzednich zajęć. Funkcje związane z obsługą tych urządzeń (LED, BUZZER, Klawiatura) muszą być realizowane w przerwaniu generowanym przez timer (dla zapewnienia jakości obsługi). Warto nadmienić iż ich implementacja powinna być zwięzła a czas zajmowany przez podprogram obsługi przerwania jak najmniejszy, nawet kosztem funkcji interfejsu (API) - bardziej skomplikowanych. Podobnie funkcje obsługi transmisji szeregowej RS232 powinny dostarczać funkcje wiarygodnego (bez gubienia i przekłamywania znaków) przekazywania informacji (między STRC51 a resztą „świata”).

Funkcje obsługi LCD – ćwiczone w ramach zajęć poprzedniego semestru – też można opcjonalnie dołączyć. Ich działanie nie jest czasowo krytyczne (w przeciwieństwie do wyświetlacza LED) i nie wymaga dodatkowego komentarza.

4.2. Zarządzanie pamięcią.

Zasoby pamięciowe STRC51, są ubogie w większości przypadków trzeba nimi oszczędnie gospodarować. Tabela nr 1. przedstawia mapę zewnętrznej pamięci systemu.

| Adresy | Rozmiar [B] | Przeznaczenie | |
|---------------|-------------|---|---------------|
| | | Obszar CODE | Obszar XDATA |
| 0xC000-0xFFFF | 16384 | - | U10 |
| 0x8000-0xBFFF | 16384 | - | U12, U15 |
| 0x1000-0x7FFF | 28671 | Pamięć kodu i danych (Uwaga adres 0x4000 to miejsce skoku po załadowaniu programu przez monitor) | |
| 0x0000-0x0FFF | 4096 | Program monitora | Obszar danych |

Tabela 1. Mapa pamięci

Aby system stał się bardziej użyteczny powinien udostępniać implementacje dobrze przemyślanego sposobu zarządzania pamięcią.

Proponowane jest zaimplementowanie funkcji o prototypach podobnych do poniższych:

```

void *malloc(int n);      - przydzielenie pamięci o wielkości n bajtów,
void free(void* p);      - zwolnienie pamięci od adresu wskazanego przez p,
int heapsize(void);      - zwrócenie wielkości sterty (dostępnego obszaru pamięci).

```

Ważne jest aby przydział pamięci - wyszukanie odpowiedniego obszaru - trwa podobnie jak zwalnianie jak najkrócej.

5. Zegar systemowy

Zegar systemowy możliwy jest do zaimplementowania przez dodanie kodu do procedur obsługi przerwania (od przepelnienia timera) oraz wbudowanie w procedury POST dodatkowego kodu synchronizującego czas systemowy z zawartością zewnętrznego modułu RTC. Istnieje także uboższa alternatywa polegająca na zaimplementowaniu tylko procedur RTC.

Przykład liczenia sumy zgodnie z alg. CRC 8 (według wielomianu: $x^8 + x^5 + x^4 + 1$)

//tabela stałych niezbędnych podczas liczenia CRC 8

```
code unsigned char TableCRC8b[256]={
  0,   94, 188, 226, 97,   63,  221, 131, 194, 156, 126, 32,  163, 253, 31,  65,
  157, 195, 33,  127, 252, 162, 64,  30,  95,  1,   227, 189, 62,  96,  130, 220,
  35,  125, 159, 193, 66,  28,  254, 160, 225, 191, 93,  3,   128, 222, 60,  98,
  190, 224, 2,   92,  223, 129, 99,  61,  124, 34,  192, 158, 29,  67,  161, 255,
  70,  24,  250, 164, 39,  121, 155, 197, 132, 218, 56,  102, 229, 187, 89,  7,
  219, 133, 103, 57,  186, 228, 6,   88,  25,  71,  165, 251, 120, 38,  196, 154,
  101, 59,  217, 135, 4,   90,  184, 230, 167, 249, 27,  69,  198, 152, 122, 36,
  248, 166, 68,  26,  153, 199, 37,  123, 58,  100, 134, 216, 91,  5,   231, 185,
  140, 210, 48,  110, 237, 179, 81,  15,  78,  16,  242, 172, 47,  113, 147, 205,
  17,  79,  173, 243, 112, 46,  204, 146, 211, 141, 111, 49,  178, 236, 14,  80,
  175, 241, 19,  77,  206, 144, 114, 44,  109, 51,  209, 143, 12,  82,  176, 238,
  50,  108, 142, 208, 83,  13,  239, 177, 240, 174, 76,  18,  145, 207, 45,  115,
  202, 148, 118, 40,  171, 245, 23,  73,  8,   86,  180, 234, 105, 55,  213, 139,
  87,  9,   235, 181, 54,  104, 138, 212, 149, 203, 41,  119, 244, 170, 72,  22,
  233, 183, 85,  11,  136, 214, 52,  106, 43,  117, 151, 201, 74,  20,  246, 168,
  116, 42,  200, 150, 21,  75,  169, 247, 182, 232, 10,  84,  215, 137, 107, 53
};
unsigned char crc8(const void *buf, unsigned char len){
  unsigned char counter;
  unsigned char crc=0;
  for(counter=0; counter<len; counter++){
    crc=TableCRC8b[crc ^ ((char*)buf)[counter]];
  }
  return crc;
}
```

Przykład liczenia sumy zgodnie z alg. CRC 16 (według wielomianu: $x^{16} + x^{15} + x^2 + 1$)

//tabela stałych niezbędnych podczas liczenia CRC 16

```
code unsigned short crc16tab[256]={
  0x0000,0x1021,0x2042,0x3063,0x4084,0x50a5,0x60c6,0x70e7,0x8108,0x9129,0xa14a,0xb16b,
  0xc18c,0xd1ad,0xe1ce,0xf1ef,0x1231,0x2252,0x3273,0x4294,0x52b5,0x62d6,0x72f7,0x8318,
  0x9339,0xa35a,0xb37b,0xc39c,0xd3bd,0xe3de,0xf3ff,0x1443,0x2464,0x3485,0x44a6,0x54c7,
  0x64e8,0x7509,0x852a,0x954b,0xa56c,0xb58d,0xc5ae,0xd5cf,0xe5ef,0xf610,0x16d1,0x26f2,
  0x3713,0x4734,0x5755,0x6776,0x7797,0x87b8,0x97d9,0xa7fa,0xb81b,0xc83c,0xd85d,
  0xe87e,0xf89f,0x1941,0x2962,0x3983,0x49a4,0x59c5,0x69e6,0x7a07,0x8a28,0x9a49,0xaa6a,
  0xba8b,0xcab9,0xdad0,0xeae1,0xfaf2,0x1b55,0x2b76,0x3b97,0x4bb8,0x5bd9,0x6bf0,0x7c11,
  0x8c32,0x9c53,0xac74,0xbc95,0xcdb6,0xddb7,0xedd8,0xfdf9,0x1e13,0x2e34,0x3e55,0x4e76,
  0x5e97,0x6eb8,0x7ed9,0x8ef0,0x9f11,0xaf32,0xbf53,0xcf74,0xdf95,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x6ead,
  0x7eae,0x8eef,0x9f10,0xaf31,0xbf52,0xcf73,0xdf94,0xedb5,0xfdd6,0x1008,0x2029,0x304a,
  0x406b,0x508c,0x60ad,0x70ce,0x80ef,0x9110,0xa131,0xb152,0xc173,0xd194,0xe1b5,0xf1d6,
  0x1208,0x2229,0x324a,0x426b,0x528c,0x62ad,0x72ce,0x82ef,0x9310,0xa331,0xb352,0xc373,
  0xd394,0xe3b5,0xf3d6,0x1408,0x2429,0x344a,0x446b,0x548c,0x64ad,0x74ce,0x84ef,0x9510,
  0xa531,0xb552,0xc573,0xd594,0xe5b5,0xf5d6,0x1608,0x2629,0x364a,0x466b,0x568c,0x66ad,
  0x76ce,0x86ef,0x9710,0xa731,0xb752,0xc773,0xd794,0xe7b5,0xf7d6,0x1808,0x2829,0x384a,
  0x486b,0x588c,0x68ad,0x78ce,0x88ef,0x9910,0xa931,0xb952,0xc973,0xd994,0xe9b5,0xf9d6,
  0x1a08,0x2a29,0x3a4a,0x4a6b,0x5a8c,0x6aad,0x7ace,0x8aef,0x9b10,0xab31,0xbb52,0xcbb7,
  0xdb98,0xebb9,0xfbd0,0x1c08,0x2c29,0x3c4a,0x4c6b,0x5c8c,0x6cad,0x7cae,0x8cef,0x9d10,
  0xad31,0xbd52,0xcd73,0xdd94,0xedb5,0xfdd6,0x1e08,0x2e29,0x3e4a,0x4e6b,0x5e8c,0x
```