

SDC_One – komputer zdefiniowany programowo z klasycznym mikroprocesorem (3)

Komputer docelowy

W pierwszym dwóch artykułach serii przedstawiliśmy ogólną koncepcję komputera zdefiniowanego programowo oraz szczegóły implementacji przez mikrokontroler STM32 krytycznych czasowo operacji związanych z realizacją protokołu szyny mikroprocesora. W trzeciej części cyklu, przedstawimy zasoby dostępne dla komputera docelowego – pamięć i wejście-wyjście, zrealizowane przez sprzęt i oprogramowanie mikrokontrolera.

Docelowy komputer składa się z mikroprocesora oraz ze zrealizowanych przez mikrokontroler STM32L4 pamięci i wielofunkcyjnego sterownika wejścia-wyjścia Mini_IO. Pamięć dostępna dla mikroprocesora w podstawowej wersji SDC_One ma rozmiar 64 KiB. Cała pamięć może być odczytywana i zapisywana przez procesor, więc zachowuje się ona jak pamięć RAM. Przy uruchomieniu SDC_One do pamięci jest ładowany, w zależności od typu użytego procesora, program ładujący system operacyjny CP/M lub program demonstracyjny.

Mini_IO

Sterownik wejścia-wyjścia Mini_IO umożliwia współpracę procesora komputera SDC_One z urządzeniami wejścia-wyjścia, na które składają się:

- dioda LED RGB na płycie mikrokontrolera i dioda LED na płycie Nucleo64, sterowane z wyjść PWM timerów mikrokontrolera STM32L4,
- przycisk umieszczony na płycie Nucleo,
- konsola umożliwiająca dwukierunkowe połączenie z PC poprzez wirtualny port szeregowy,
- pamięć masowa w postaci dwóch jednostek dyskowych, symulowanych w pamięci Flash i RAM mikrokontrolera.

Ponadto, Mini_IO zawiera prosty timer systemowy, który może odmierzać odcinki czasu o długości ustawionej przez oprogramowanie komputera docelowego, w zakresie od 1 ms do ponad 1 s.

Wszystkie urządzenia zostały zrealizowane przy użyciu zasobów sprzętowych i oprogramowania mikrokontrolera STM32. Sterownik Mini_IO jest dostępny dla mikroprocesora poprzez 16 8-bitowych rejestrów odwzorowanych w przestrzeni wejścia-wyjścia, a jeśli procesor zastosowany w komputerze takiej nie posiada – w 16 lokacji przestrzeni adresowej pamięci. Nazwy i opis rejestrów Mini_IO przedstawiono poniżej.

Sterowanie bitowe LED

Rejestr IO_LED służy do prostego sterowania diod LED (zaświecanie, gaszenie). Przy zapisie rejestru pole bitowe FUN określa operację, a pole LED – wartość sterującą stanem diod. W zależności od wartości pola FUN, pojedynczy zapis może na różne sposoby zmienić stan bitów sterujących świeceniem wybranych diod.

- FUN = 00 – ustawienie stanu diod wg. wartości zapisanej do pola LED.
- FUN = 01 – ustawienie w stan 1 bitów, które w polu LED mają wartość 1.
- FUN = 10 – ustawienie w stan 0 bitów, które w polu LED mają wartość 1.
- FUN = 11 – zmiana stanu (negacja) bitów, które w polu LED mają wartość 1.

Jeżeli bit sterujący stanem diody zostanie wyzerowany – dioda gaśnie, a ustawienie powoduje zaświecenie diody z pełną jasnością.

Konsola i timer

Do współpracy z konsolą i timerem oraz odczytem stanu przycisku służą trzy rejestry Mini_IO:

1. Rejestr IO_CON służy do zapisu danych przesyłanych do konsoli i odczytu danych odbieranych z konsoli (terminala). Zawiera on 8-bitową daną przesłaną lub odebraną przez interfejs VCOM konsoli.
2. Rejestr IO_TIM zawiera okres timera systemowego. Jego wartość początkowa wynosi 0, co odpowiada zatrzymaniu timera. Bit 7 – G – określa granularność okresu timera. Wartość 0 odpowiada jednostkom 1 ms, 1 – 8 ms. Bity 6...0 zawierają okres timera wyrażony w jednostkach zależnych od wartości bitu G. Zapis wartości 0 powoduje zatrzymanie timera i wyzerowanie znacznika końca okresu w rejestrze IO:CSR. Zapis wartości z zakresu 1...127 powoduje uruchomienie timera. Maksymalny okres timera, odpowiadający wartości 0xff rejestru IO_TIM, może trwać 1016 ms.
3. Rejestr IO_CSR zawiera znaczniku stanu timera i interfejsu konsoli:
 - Bit 0 – RXNE – ma wartość 1, gdy w rejestrze IO_CON jest dostępny do odczytu znak przesłany z konsoli. Bit ten jest zerowany przy odczycie rejestru IO_CON i ustalany przy odebraniu znaku z konsoli.
 - Bit 1 – TXE – ma wartość 1, gdy interfejs konsoli jest gotowy do wysłania znaku na konsolę (do rejestru IO_CON można zapisać kolejny znak). Bit ten jest zerowany przy zapisie rejestru IO_CON, po czym jest on samoczynnie

	7	6	5	4	3	2	1	0
LED			FUN		LEDN	LEDB	LEDG	LEDR
	r:RD w:TD							
	DIV8	PERIOD						
	BTN				TIF		TXE	RXNE
DUTYR	DUTYR							
DUTYB	DUTYB							
DUTYG	DUTYG							
DUTYN	DUTYN							
MS_SEC	Sector							
MS_TRK	Track							
MS_UNIT	Unit							
MS_CMD	r:STATUS w:CMD							
MS_A0	BufAddr[7..0]							
MS_A1	BufAddr[15..8]							
MS_A2	BufAddr[23..16]							
	reserved							

Rysunek 1. Rejestry sterownika wejścia-wyjścia Mini_IO

ustawiany w stan 1, gdy interfejs staje się gotowy na przyjęcie kolejnego znaku.

- Bit 2 – TIF jest ustawiany po zakończeniu okresu timera. Jest on zerowany poprzez zapis do rejestru IO_CSR wartości, w której bit ten jest jedynką.

Sterowanie jasnością LED

Rejestry IO_DUTYR, IO_DUTYRG, IO_DUTYB i IO_DUTYN umożliwiają sterowanie jasnością świecenia diody RGB i diody LED zamontowanej na płycie Nucleo64. Wartość zapisywana do tych rejestrów jest interpretowana jako licznik ułamka o mianowniku 250, określającego wypełnienie przebiegu sterującego jasnością diody. Ustawienie bitu w rejestrze IO_LEDS ma skutek taki sam, jak zapis wartości 255 do rejestru wypełnienia. Wyzerowanie bitu w rejestrze IO_LEDS ma skutek taki sam, jak zapis wartości 0 do rejestru wypełnienia. Zawartości rejestrów wypełnień i rejestru IO_LEDS nie są ze sobą powiązane – przy odczycie jest zawsze pobierana wartość wpisana ostatnio do odczytywanego rejestru.

Sterownik pamięci masowej

Sterownik pamięci masowej emuluje dwie jednostki dyskowe. Pierwsza z nich korzysta z pamięci Flash mikrokontrolera sterującego, a druga – z pamięci RAM. Pierwsza jednostka ma pojemność 512 kB, a druga – 32 kB. Zawartość drugiej jednostki jest tracona przy wyłączeniu komputera, można ją jednak przed wyłączeniem skopiować do pamięci Flash mikrokontrolera używając polecenia monitora. Po włączeniu SDC_One zawartość jest inicjowana z pamięci Flash.

Emulowane jednostki pamięci masowej mają ścieżki o pojemności 2 kB. Dla systemów CP/M ścieżka składa się z 16 sektorów po 128 bajtów, Pierwsza jednostka ma 256 ścieżek, a druga 16.

Sterownik jest widziany przez komputer docelowy jako jako 7 rejestrów 8-bitowych.

IO_MSSEC zawiera numer sektora dla operacji odczytu lub zapisu. Dozwolony zakres wartości od 0 do 15.

IO_MSTRK zawiera numer ścieżki. Zakres zależy od numeru jednostki: 0...255 dla jednostki 0, 0...15 dla jednostki 1.

IO_MSUNIT zawiera numer jednostki – 0 lub 1.

Zapis do rejestru IO_MSCMD powoduje wykonanie operacji zależnej od zapisywanej wartości:

- 1 – odczyt sektora z jednostki pamięci masowej do pamięci komputera,
- 2 – zapis sektora z pamięci komputera do jednostki pamięci masowej,
- 3 – synchronizacja wybranej jednostki,
- 4 – synchronizacja wszystkich jednostek.

Synchronizacja jednostki 0 polega na skopiowaniu zawartości buforów ostatnio używanych sektorów do pamięci Flash mikrokontrolera. Synchronizacja jednostki 1 polega na przepisaniu całej zawartości jednostki z pamięci RAM do pamięci Flash.

Odczyt rejestru IO_MSCMD zwraca kod zakończenia operacji. Wartość 0 oznacza pomyślne zakończenie; wartość niezerowa – błąd. Odczyt każdego z pozostałych rejestrów sterownika zwraca ostatnio zapisaną wartość.

W celu odczytu lub zapisu sektora przez komputer docelowy, jego oprogramowanie powinno:

- zapisać numer sektora do rejestru IO_MSSEC,
- zapisać numer ścieżki do rejestru IO_MSTRK,
- zapisać numer jednostki do rejestru IO_MSUNIT,
- zapisać adres bufora sektora do rejestrów IO_MSADDR0, IO_MSADDR1 i IO_MSADDR2,
- zapisać polecenie do rejestru IO_MSCMD.

W wersjach SDC_One z mikroprocesorami 80C85 i Z80CPU sterownik Mini_IO jest odwzorowany w przestrzeni adresowej wejścia-wyjścia. W wersjach komputera wyposażonych w procesory bez wydzielonej przestrzeni wejścia-wyjścia (MC68008 i 65C02) sterownik jest dostępny dla procesora w przestrzeni adresowej pamięci. Domyślny adres bazowy sterownika (0 dla 80C85, Z80CPU i 65C02, 0xFFFF0 dla MC68008) może być zmieniany poleceniem monitora systemu.

W kolejnej części zaprezentujemy schematy ideowe SDC_One oraz szczegóły połączeń pomiędzy mikrokontrolerem STM32L4 i procesorem komputera docelowego.

Julia Kosowska
Grzegorz Mazur