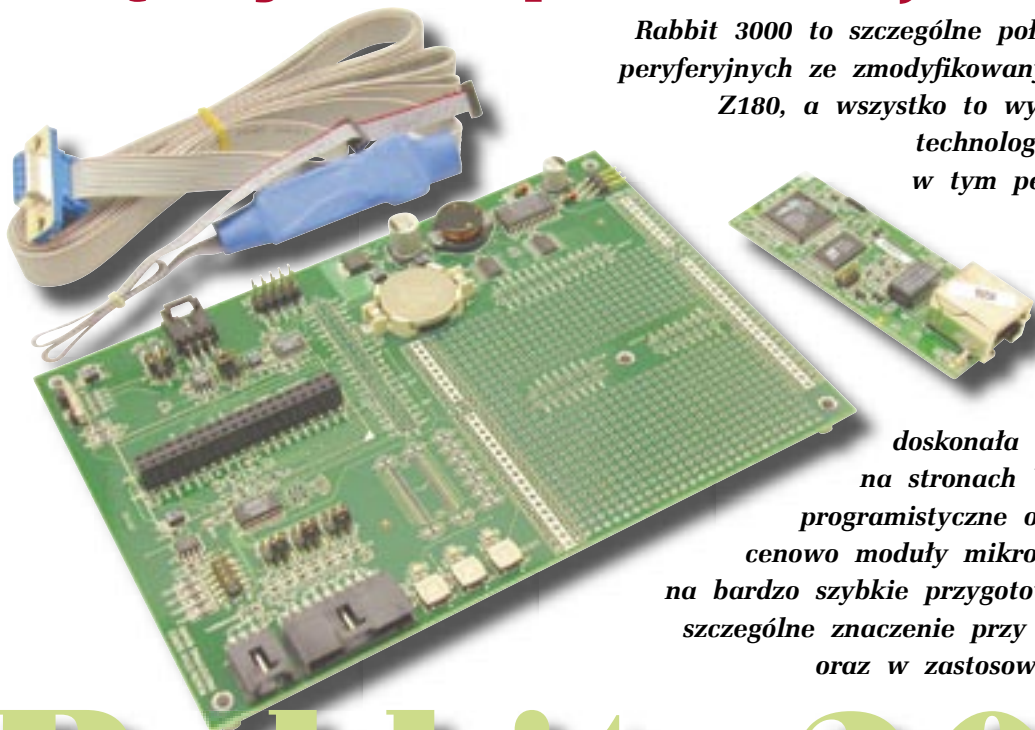


Więcej niż procesor, część 1

Rabbit 3000 to szczególne połączenie wielu układów peryferyjnych ze zmodyfikowanym rdzeniem procesora Z180, a wszystko to wykonane w nowoczesnej technologii 0.35 μm . Nie byłoby w tym pewnie nic szczególnego, gdyby nie specyficzny klimat, który udało się stworzyć producentowi wokół tych urządzeń. Składają się na to: doskonała dokumentacja dostępna na stronach WWW, dobre narzędzia programistyczne oraz gotowe, atrakcyjne cenowo moduły mikrokontrolera pozwalające na bardzo szybkie przygotowanie aplikacji, co ma szczególne znaczenie przy produkcji niskoseryjnej oraz w zastosowaniach hobbystycznych.



Rabbit 3000

Atrakcyjność wyrobu zdecydowanie podnosi obsługa protokołów TCP/IP. Lista układów peryferyjnych Rabbita 3000 jest imponująca, a szczegóły ich obsługi mogą wprawić w zakłopotanie. W praktyce nie musimy się jednak tym wszystkim martwić, ponieważ dostarczany kompilator języka C obsługuje wszystkie zasoby mikrokontrolera za pomocą gotowych funkcji bibliotecznych. Obecnie jest dostępna już druga generacja układów Rabbita. Spróbujmy przyjrzeć się wewnętrznym zasobom tego mikrokontrolera.

Rdzeń Z180

Jak zostało wspomniane, Rabbit 3000 został zbudowany na bazie zmodyfikowanego rdzenia procesora Z180, będącego rozwinięciem Z80. Celem każdej modyfikacji jest ulepszenie pierwowzoru. Skutkiem ubocznym poczynionych poprawek jest brak możliwości wykorzystania narzędzi od Z180. Niektóre instrukcje zostały po prostu usunięte, inne zmienione. Zrezygnowano również z metody bezpośredniego dostępu do pamięci (DMA). Rabbit 3000 ma więc własne narzędzia

programistyczne uwzględniające wszystkie jego specyficzne aspekty. Można się zastanawiać, dlaczego wybrano rdzeń Z180 zamiast np. nowoczesnego rdzenia RISC-owego. Odpowiedź jest prozaiczna: firma Z-World bazuje na wieloletnim doświadczeniu zdobytym podczas produkcji systemów mikroprocesorowych z procesorem Z180. Wykorzystano między innymi istniejące narzędzia programistyczne, poza tym człowiek łatwo się przyzwyczaja, a procesory z linii 8080/Z80/180 mają ugruntowaną, ponad 20-letnią tradycję. Jeśli dodamy do tego wydajny kompilator języka C, to wcale nie będą nam przeszkadzały nieco długie cykle maszynowe czy wymyślne tryby adresowania.

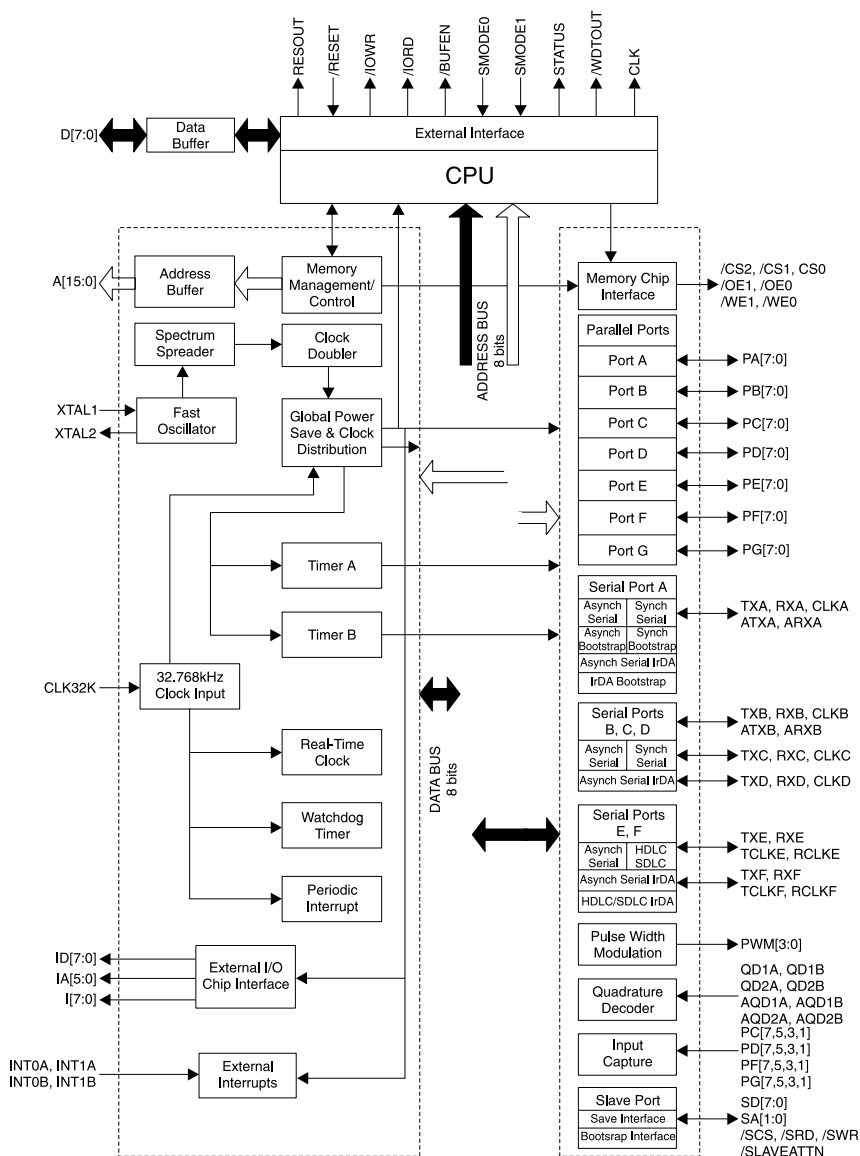
Zasilanie

Zastosowanie nowych technologii pozwoliło na zasilanie układu napięciem z zakresu od 1,8 do 3,6 V. Większość wejść toleruje napięcie 5 V, co umożliwia współpracę z 5-woltowym otoczeniem HCT CMOS. Układ pobiera ze źródła zasilania prąd o natężeniu 65 mA przy częstotliwości oscylatora

równiej 30 MHz i napięciu zasilania 3,3 V. Pobór prądu jest proporcjonalny do częstotliwości taktowania i napięcia zasilającego. Dla napięcia 1,8 V i częstotliwości 1 MHz wynosi ok. 1 mA i może być obniżony do poziomu 20 μA przy taktowaniu sygnałem 32,768 kHz. Warto również zwrócić uwagę na szeroki zakres temperatury pracy od -55°C do $+85^{\circ}\text{C}$. Na **rys. 1** przedstawiono schemat blokowy Rabbita 3000.

Taktowanie

Sygnał taktujący rdzeń procesora otrzymywany jest z głównego oscylatora współpracującego z zewnętrznym rezonatorem kwarcowym lub ceramicznym z zakresu od 1,8 MHz do 25 MHz. Częstotliwość ta może zostać wewnętrznie podwojona (50 MHz) lub podzielona przez 2, 4, 6 lub 8. Możliwe jest też taktowanie rdzenia zewnętrznym sygnałem 32,768 kHz, co pozwala na obniżenie poboru prądu do poziomu ok. 20 μA . Sygnał ten steruje również pracą układu *watch-doga* oraz wyznacza prędkość bodową kanału szeregowego A podczas tzw. „zimnego startu”. Prędkość taktowa-



Rys. 1 Schemat blokowy Rabbita 3000

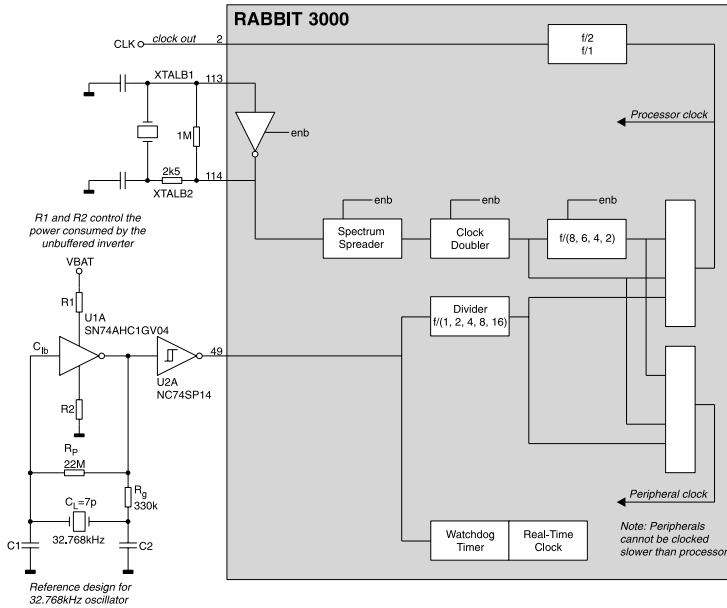
nia może być zmieniana „w locie”, co znajduje zastosowanie w urządzeniach zasilanych bateryjnie. Na rys. 2 przedstawiony jest system rozprowadzania sygnału zegarowego w układzie Rabbit 3000.

Przestrzeń adresowa

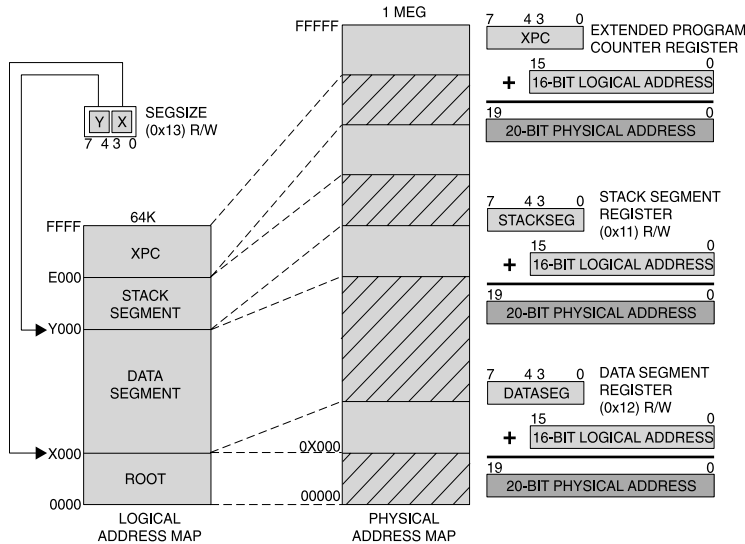
Rabbit 3000 umożliwia fizyczny dostęp do 1 MB pamięci (20 linii adresowych). Sam procesor „widzi” jednak tylko 64 kB pamięci logicznej. Odwzorowaniem pamięci logicznej w fizyczną zajmuje się układ MMU (*Memory Mapping Unit*). Obszar 64 kB pamięci logicznej podzielony

jest na 4 części: *Root*, *Data*, *Stack*, *XPC*. Każda z nich może być odwzorowana w różne regiony pamięci fizycznej. Wyjątkiem jest segment *Root*, który zawsze odwzorowany jest w najniższy fragment pamięci fizycznej. Jeśli korzystamy z kompilatora DynamicC, mapowanie pamięci odbywa się w sposób niewidoczny dla użytkownika. Umożliwia to tworzenie aplikacji, których kod przekracza 64 kB. Niestety kompilator nie potrafi wykorzystać dużej przestrzeni adresowej dla alokacji zmiennych statycznych. W rezultacie pozostaje do tego celu jedynie 20-40 kB – o wiele za mało dla niektórych rodzajów aplikacji. W takim przypadku pozostaje jedynie korzystanie z funkcji transferu danych między pamięcią logiczną a fizyczną. Zasady mapowania pamięci wyjaśnia rys. 3.

Rabbit 3000 – 8-bitowy mikrokontroler o wydajności porównywalnej z układami 16-bitowymi.



Rys. 2. System rozprowadzania sygnału zegarowego



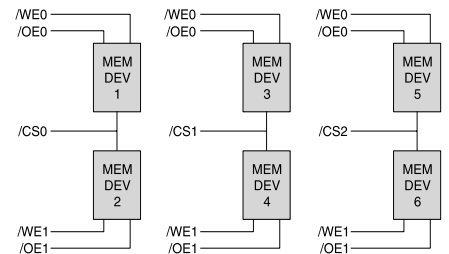
Rys. 3. Mapowanie pamięci

Interfejs pamięci

Układ współpracuje z pamięcią statyczną za pośrednictwem 8-bitowej

magistrali danych, 20-bitowej magistrali adresowej oraz kilku konfigurowalnych linii sterujących. Cykle

POSSIBLE COMBINATIONS OF MEMORY CONTROL SIGNALS



Rys. 4. Interfejs pamięci

dostępu do pamięci mogą być sprzętowo wydłużane (*wait states*) w celu zapewnienia poprawnej współpracy z wolniejszymi układami. Możliwe jest podłączenie 6 układów pamięci bez stosowania dodatkowych układów dekodujących (*glueless interface*). Typowa konfiguracja to jeden układ pamięci *Flash* oraz jedna „kostka” pamięci *RAM*. Rabbit 3000 posiada osobną przestrzeń adresową dla urządzeń *we/wy* wielkości 64 kB. Przestrzeń ta posiada osobne instrukcje dostępu oraz osobne wyprowadzenia sterujące */IORD*, */IOWR*. Dodatkowe sygnały strobojące o programowo ustawianej funkcji są dostępne dla wyprowadzeń portu E. Magistrale adresowa i danych mogą być wspólne z układami pamięci. Istnieje jednak możliwość wykorzystania osobnej magistrali danych i adresowej (skróconej w tym przypadku do 6 bitów) kosztem zmniejszenia liczby wyprowadzeń *we/wy*. Rozwiązanie takie upraszcza projekt płytki drukowanej oraz wpływa na obniżenie emisji zakłóceń elektromagnetycznych, ponieważ dodatkowa magistrala *we/wy* jest aktywna tylko podczas operacji *we/wy*. Interfejs pamięci jest przedstawiony na **rys. 4**.

Daniel Pilecki-Silva