

Wirtualne peryferia

SoC - jak to się robi w praktyce?



O problemach, jakie stają przed projektantem układów SoC (*System-on-a-Chip*), a zwłaszcza o trudnościach w znalezieniu gotowych komponentów, które oszczędzą trudu wykonywania całej pracy projektowej od początku, pozwalając oprzeć się na sprawdzonych i wypróbowanych rozwiązaniach, pisaliśmy w EP10/2002. Lekarstwem na bolączki tego typu jest działalność firm zajmujących się dostarczaniem wirtualnych komponentów.

Jedną z takich firm jest Evatronix S.A. W artykule przedstawimy dwa wirtualne komponenty opracowane przez inżynierów z tej firmy. Wśród prezentowanych komponentów są: ośmiobitowy

O różnych wariantach układów SoC piszemy w EP od dłuższego czasu. Szczególnie jasną przyszłość można wróżyć implementacjom „systemów” na krzemie w układach PLD i ASIC, ponieważ coraz większe grono producentów tworzy wirtualne komponenty, dzięki którym realizacja nawet bardzo skomplikowanego projektu jest stosunkowo łatwa.

mikrokontroler oraz współpracujący z nim zmiennoprzecinkowy układ arytmetyczny.

R80515 - szybki mikrokontroler 8-bitowy

Dla aplikacji wymagających popularnego, elastycznego i przy tym wydajnego mikro-

kontrolera ośmiobitowego firma Evatronix S.A. opracowała moduł szybkiego mikrokontrolera, oznaczony symbolem R80515. Jest on funkcjonalnie w pełni zgodny z intelowskim pierwowzorem 80C31, dzięki czemu mogą z niego korzystać szerokie rzesze konstruktorów zaznajomionych z tą popular-

Wirtualny komponent - co to jest?

Wszyscy bez wyjątku producenci wirtualnych komponentów dostarczają je na rynek w postaci syntezowalnych opisów w językach HDL (Hardware Description Language). Po ich skompilowaniu za pomocą łatwo dostępnych programów narzędziowych EDA, odpowiednio są konfigurowane docelowe (już krzemowe) układy PLD.

ną architekturą. Litera R w nazwie komponentu oznacza, że jądro mikrokontrolera R80515 zawiera rozwiązania zapożyczone z architektury RISC. W R80515 zastosowano m.in. przetwarzanie potokowe, zredukowano liczbę cykli potrzebnych do wykonywania rozkazów, zaimplementowano także sprzętową realizację funkcji arytmetycznych oraz sterownik pamięci mogący współpracować z pamięciami synchronicznymi i asynchronicznymi.

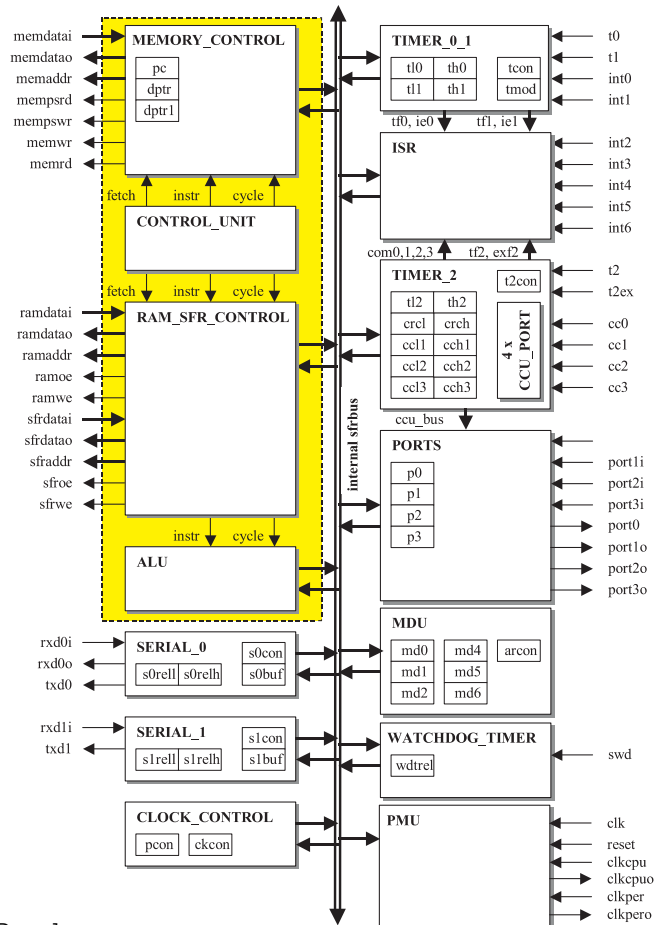
Schemat blokowy mikrokontrolera R80515 pokazano na rys. 1. Rdzeń mikrokontrolera, oznaczony na rys. 1 żółtym prostokątem, składa się z jednostki sterowania (*Control_unit*) zawierającej dekodery rozkazów, jednostki wykonawczej (ALU) realizującej ośmiobitowe operacje arytmetyczne, logiczne i rotacje, oraz jednostki sterowania pamięcią (*Ram_control*), mogącej obsłużyć przestrzeń wewnętrznej pamięci danych sięgającą 256 bajtów oraz zewnętrzne pamięci danych i programu sięgające 64 kB każda. Dodatkowo w rdzeniu procesora zaimplementowana jest jednostka sterowania rejestrami (*Ram_sfr_control*).

Poza rdzeniem znajduje się szesnastobitowy układ arytmetyczny (MDU) operujący na liczbach całkowitych, który realizuje między innymi operacje mnożenia i dzielenia, a także operacje logiczne. Również poza rdzeniem znajduje się układ przerwań (*Isr*) obsługujący trzynaście źródeł przerwań zorganizowanych w cztery priorytety uprzywilejowania, oraz układ watchdog (*Watchdog_timer*) czuwający nad prawidłowym wykonywaniem programu. Mikrokontroler posiada wbudowane porty wejść i wyjść równoległych i szeregowych. Blok portów równoległych (*Ports*) obsługuje cztery równoległe dwukierunkowe porty wejścia

wyjścia, stosowane również do dostępu do zewnętrznej pamięci, w trybie multipleksowania magistrali adresowej. Porty szeregowo realizowane są przez dwa układy obsługi portów szeregowych (*Serial_0* i *Serial_1*), mogące pracować dwukierunkowo (*full duplex*). Port pierwszy i drugi mogą pracować w trybie UART z ramką ośmiobitową oraz dziewięciobitową. Dodatkowo port pierwszy może pracować w trybie synchronicznym. Uzupełnieniem dostępnych peryferiów zaimplementowanych w komponencie są trzy szesnastobitowe układy czasowo-licznikowe (*Timer_0_1*, *Timer_2*), pracujące (oprócz trybów znanych z klasycznego 8051) dodatkowo w trybie generowania sygnału PWM (*Compare*), oraz w trybie pomiaru szerokości impulsu (*Capture*).

Komponent R80515 może pracować w trybie obniżonego poboru mocy lub w trybie całkowitego wyłączenia poboru mocy. Tryby te (IDLE i STOP) są sterowane przez jednostkę PMU.

Dzięki elastyczności cechującej technologię wirtualnych komponentów, klient nabywający R80515 może zmodyfikować jego podstawowe właściwości. Przykładowo, magistrale adresowe zamiast magistral danych mogą być współdzielone z wyjściami portów lub też mogą być wyprowadzone osobno. Również porty szeregowo zamiast korzystać z wyprowadzeń portu trzeciego mogą być wyprowadzone osobno. Istnieje możliwość łatwej modyfikacji cyklu dostępu do pamięci programu i danych w celu dostosowania



Rys. 1

procesora do różnych układów pamięci i wejścia wyjścia.

Badania oraz testy aplikacyjne przeprowadzona przez Dział Klienta Wewnętrznej firmy Evatronix S.A., bazujące na porównaniu wyników otrzymanych z testów Dhrystone 1.1 dla oryginalnego 80C31 i R80515 wykazały 6,8-krotną przewagę wydajnościową komponentu nad oryginałem. Wyniki otrzymano dla tej samej częstotliwości taktowania, na tym samym systemie uruchomieniowym Evix Board 3, kompilując kod testu kompilatorem firmy Tasking. Otrzymane wyniki w teście Dhrystone 1.1 dla procesora Intel 80C51 taktowanego zegarem 12 MHz to 198 Dhrystone/s, natomiast taktowany tym samym zegarem procesor R80515 osiąga 1353 (tysiąc trzysta pięćdzie-

siat trzy) Dhrystone/s. Dla porównania, system IBM PC/AT 80286 taktowany zegarem 6 MHz uzyskuje w tym samym teście skompilowanym kompilatorem C firmy Microsoft wynik 1388 Dhrystone/s.

Możliwości R80515 choć imponujące pod względem przetwarzania stałoprzecinkowego mogą nie zadowalać użytkownika planującego zastosować ten procesor do intensywnego przetwarzania zmiennoprzecinkowego. W sukurs przychodzi moduł stałoprzecinkowej arytmetyki szesnastobitowej (MDU), zgodny ze standardem Siemens 80C515. Większość nowoczesnych kompilatorów języka C potrafi wykorzystać obecność tego modułu, w celu przyspieszenia realizacji obliczeń numerycznych. Przyspieszenie działań zmiennoprzecinkowych zbadano przy pomocy testu Whetstone A001. Wyniki jakie otrzymano dla 8051 przy 6 MHz wyniosły 11,3 Whetstone/s, natomiast przy tym samym zegarze dla R80515 otrzymano 88 Whetstone/s.

Różnice pomiędzy wartościami

Sposoby „kodowania” wirtualnych komponentów

Najczęściej wirtualne komponenty są opisywane za pomocą języków: VHDL, Verilog, spotykane są także opisy w postaci list połączeń EDIF.

Korzyści wynikające ze stosowania wirtualnych peryferiów

Projektant układu SoC korzystający z wirtualnych peryferiów buduje swój projekt niemal klasycznie: bierze "z półki" gotowe komponenty (wirtualne), łączy je za pomocą specjalnego edytora w funkcjonalną całość, następnie kompiluje w wyniku czego otrzymuje w krótkim czasie pełnowartościowy układ o specyficznych funkcjach.

cią współczynnika przyspieszenia wykonywania testu stałoprzecinkowego (6,8 raza) i testu zmiennoprzecinkowego (7,8 raza) należy tłumaczyć odmiennym charakterem kodu programu testowego. Kod testu Whetstone zawiera bowiem w stosunku do kodu testu Dhrystone więcej rozkazów realizowanych w jednym cyklu rozkazowym. Rozkazy takie dzięki poprawionej architekturze R80515 realizowane są w jednym taktie zegarowym. Im więcej takich krótkich jednobajtowych rozkazów zawiera program tym większe jest przyspieszenie uzyskiwane przez R80515 względem klasycznego mikrokontrolera.

Ostatnim, ale nie mniej ważnym, czynnikiem wpływającym na końcową wydajność układu jest technologia, w której zostanie on fizycznie zaimplementowany. Współczesne układy FPGA i ASIC pozwalają taktować wirtualny komponent zegarami rzędu setek MHz, co w sposób oczywisty wpływa na uzyskiwane rezultaty.

Sprzętowa akceleracja obliczeń zmiennoprzecinkowych

Istnieje wiele zastosowań wymagających połączenia łatwości stosowania operacji na pojedynczych bitach z wysoką wydajnością zmiennoprzecinkową. Z tego właśnie powodu komponent R80515 wyposażono w specjalny 32-bitowy zmiennoprzecinkowy moduł (dostępny także w postaci osobnego komponentu) *AddMull 32*, realizujący sprzętowo

wo mnożenie, dodawanie i odejmowanie. Realizuje on szybkie obliczenia zmiennoprzecinkowe, operując na liczbach formatu zmiennoprzecinkowego pojedynczej precyzji (32 bity). Obliczenia wykonywane są poprzez zapis argumentów wybranego działania arytmetycznego do odpowiednich rejestrów komponentu oraz kodu określającego rodzaj działania do rejestru sterującego.

W przypadku współpracy z rdzeniem komponentu R80515 rejestry układu *AddMull* są zamapowane w obszarze rejestrów specjalnych (SFR). Komponent jest taktowany zegarem procesora, a wykonanie każdego z działań zajmuje pięć cykli zegarowych, wobec czego po zapisie słowa sterującego procesor może realizować inne zadania. Po pięciu taktach od rozpoczęcia obliczeń procesor może pobrać wynik działania z rejestru wyjściowego *AddMull*.

AddMull 32 wykonuje następujące operacje zmiennoprzecinkowe: dodawanie, odejmowanie, mnożenie, dzielenie i dodawanie w jednej operacji oraz mnożenie i odejmowanie w jednej operacji.

Zestaw działań oraz rejestrów komponentu *AddMull 32* dobrano w taki sposób, aby umożliwić optymalne wykorzystanie go do obliczeń numerycznych. Znakomitą większość spotykanych w nauce i technice funkcji matematycznych można przybliżyć w wybranym przedziale przy pomocy wielomianu przekształconego dla optymalizacji schematem Hoernera. Tak przekształcony wielomian składa się z kilku wykonywanych cyklicznie operacji mnożenia dwóch argumentów i następnie dodawania lub odejmowania od otrzymanego wyniku trzeciego argumentu. Przykładowo poniżej przed-



stawiono wielomian aproksymujący funkcję sinus w przedziale od 0 do $\pi/2$:

$$\sin(x) = (((((x^2 * a11 + a9) * X^2 + a7) * x^2 + a5) * x^2 + a3) * x^2 + a1) * x$$

a11 = -2,052108e-8
 a9 = 2,557319e-6
 a7 = -0,001987
 a5 = 0,083333
 a3 = -0,666667
 a1 = 1,;

Definiując rejestry komponentu AddMull 32 tak, aby rejestr wynikowy mnożenia był jednocześnie jednym ze argumentów dodawania redukuje się liczbę przesłań danych podczas wyliczania przedstawionego wielomianu.

Można pokusić się o porównanie wydajności kontrolera R80515 współpracującego z komponentem AddMull 32, realizującego przedstawiony powyżej algorytm, a wydajnością kontrolera pozbawio-

nego wsparcia sprzętowego, na którym realizowany jest algorytm funkcji sinus zaimplementowany w jednym z popularnych narzędzi programistycznych. Załóżmy przy tym, że dane do obliczeń znajdują w pamięci zewnętrznej systemu. Pobranie bajtu z pamięci zewnętrznej do rejestru roboczego procesora trwa dwa cykle rozkazowe, skopiowanie z rejestru roboczego do rejestru specjalnego SFR jednego bajtu potrwa jeden cykl zegarowy. W celu obliczenia podanego wielomianu należy wykonać dziesięć operacji transferu trzydziestodwubitowej danej pomiędzy pamięcią zewnętrzną a przestrzenią SFR, każdy taki transfer trwa dwanaście cykli rozkazowych. Należy przyjąć przy tym że obliczona wartość X^2 przechowywana będzie w rejestrach roboczych kontrolera i stamtąd pobierana do obliczeń. Opera-

Przykładowe implementacje R80515

Dzięki zastosowaniu dość uniwersalnego sposobu opisu R80515 (język VHDL) mikrokontroler można łatwo zaimplementować w układach PLD różnych producentów. I tak przykładowo, w przypadku układu XCV200E niezbędne było 2108 komórek Slice, a maksymalna częstotliwość taktowania wyniosła 40 MHz. W przypadku zastosowania jako układ docelowy EPA20K200C projekt zajął 3928 makrokomórek i 5 bloków ESB, a maksymalna częstotliwość taktowania wynosi 30,7 MHz.

cji transferu wartości X^2 pomiędzy rejestrami roboczymi a SFR będzie pięć, każda z nich potrwa cztery cykle. Operacji mnożenia wykonanych zostanie siedem, natomiast operacji dodawania pięć. Każda taka operacja trwa sześć cykli rozkazowych (w tym jeden na zapis słowa sterującego do rejestru AddMull i pięć rozkazów NOP poświęconych na taktowanie AddMull podczas wyliczania). Ostatecznie otrzymujemy 212 cykli rozkazowych potrzebnych na wyliczenie wartości funkcji sinus przez komponent AddMull 32. Dla porównania, przytaczając za dokumentacją kompilatora firmy Keil, procesor 80515 obliczając funkcję sinus zapisaną w języku C potrzebuje od 1422 do 3048 (średnio 2519) cykli rozkazowych.

Widoczne jest na tym przykładzie, że używając tylko i wyłącznie układu sprzętowego mnożenia i dodawania liczb zmiennoprzecinkowych można znacznie przyspieszyć nawet złożone obliczenia numeryczne, bez potrzeby wyposażania systemu w komplety koprocessor numeryczny

lub zmiennoprzecinkowy procesor DSP.

Podsumowanie

Wirtualne komponenty należą do standardowego zestawu narzędzi współczesnego projektanta systemów cyfrowych. Producenci oferują w tej postaci bardzo wiele bloków funkcjonalnych, począwszy od mikroprocesorów i mikrokontrolerów, poprzez wirtualne odpowiedniki klasycznych periferiów stosowanych w systemach mikroprocesorowych, aż po zaawansowane moduły komunikacyjne. Jak na razie, korzystanie z nich przez niewielkie firmy jest utrudnione, ponieważ w przypadku realizacji projektów niskonakładowych cena komponentów dość istotnie wpływa na końcową cenę wyrobu. Taka jest jednak przyszłość elektroniki - za kilka lat układy nie dające się programować prawdopodobnie całkowicie zagniją...

Tomasz Jakóbiec, współpraca: Mirosław Bandzerewicz

Dodatkowe informacje

Dodatkowe informacje można znaleźć na stronie www.evatronix.pl.

Wirtualnie nie musi oznaczać drogo
Wirtualne komponenty dostarczane są na rynek w postaci syntetyzowalnego kodu źródłowego w języku opisu sprzętu HDL lub w postaci zsyntetyzowanych netlist. Tylko przypadku zakupu kodu HDL jest możliwe dokonywanie zmian w wirtualnym komponencie, natomiast - co ważne - zakup netlisty jest znacznie mniej kosztowny (nawet o 50%) niż kupno wersji źródłowej.
Z netlisty można zrobić użytek dzięki bezpłatnym narzędziom programowym, wersję źródłową syntetyzować można tylko przy użyciu drogiego oprogramowania do syntezy. Zazwyczaj z wersji źródłowej korzystają producenci układów ASIC, netlista jest wystarczającym rozwiązaniem dla projektantów urządzeń i rozwiązań elektronicznych.