

Souhrn zjištěných omezení při práci s překladačem C pro PicoBlaze - PCComp

Zbyněk Křivka, Rudolf Schönecker

krivka@fit.vutbr.cz, schonec@fit.vutbr.cz

zpráva projektu
Virtuální laboratoř aplikace mikroprocesorové techniky
MŠMT 2C06008

Abstrakt dokumentu

Zde je podán soupis nalezených chyb, nekorektního chování, či špatně dokumentovaných vlastností překladače C pro PicoBlaze (PCComp). Autor překladače je Ital Francesco Poderico.

Klíčová slova

PicoBlaze, překladač C

Úvod

Při konstrukci demo příkladů pro FITKit s procesorem PicoBlaze byl řídicí software psán v jazyce C. Byl použit jediný dostupný překladač C pro PicoBlaze – PCComp (autor překladače je Francesco Poderico). Při konstrukci byly registrovány chyby a nekorektní chování překladače, které ztěžovalo (v některých případech znemožňovalo) použít jinak korektní C kód.

Překladač PCComp obsahuje pouze velmi stručnou a nepřesnou dokumentaci z roku 2004 a navíc stránky projektu jsou již několik měsíců nedostupné a je pravděpodobné, že projekt jako takový byl bez budoucí podpory ukončen nebo bez informování uživatelů uzavřen před veřejností.

S cílem usnadnit příští vývoj, popř. poskytnout motivaci a vodítka pro tvorbu nového překladače C pro PicoBlaze, je zde podán soupis nalezených chyb, nekorektního chování, či špatně dokumentovaných vlastností překladače C pro PicoBlaze, na které jsme při konstrukci demo příkladů narazili.

Chyby překladače PCComp

- chyba v přetypování `int` na `byte` – vrací chybnou hodnotu – dočasné řešení spočívá ve využití operátoru `shift` – spodní byte je nutno posunout (`shift`) o nula bitů, aby vše fungovalo, jak fungovat má (viz demo `calculator.c`)

```
b1 = (byte) (keycode>>0);  
// b1 = (byte) (keycode); NEFUNGUJE !!!  
b2 = (byte) (keycode>>8);
```
- nefungují logické operátory `&&` a `||` – dočasné řešení spočívá v rozepsání testů podmínek, což je ale při velikosti architektury a výsledného kódu neúnosné
- většinou se v aritmetických operacích pracuje s 16-bitovými operandy i když to není potřeba – při velikosti architektury opět neúnosné
- z funkce nelze vracet konstantu – je nutné vytvořit pomocnou proměnnou a naplnit ji konstantou a tu pak přes proměnnou vrátit (PCComp to zavádí kvůli optimalizaci všech RETURN-příkazů podprogramu, které jsou sdruženy dohromady a umístěny na konec podprogramu)

Specifické chování

- `#asm-#endasm` blok není zpracováván, přímo se vkládá nezměněn do výsledného výstupu, proto přímo uvnitř něj
 - nelze použít**
 - další direktivy preprocesoru
 - přímo názvy lokálních parametrů
 - přímo názvy globálních proměnných
 - lze použít**
 - **globální proměnné programu 8-bitové** (`char`, `unsigned char`, `pointer`) přidáním prefixu „_“ – PCComp název s daným prefixem použije v assembleru jako konstantu specifikující adresu ve scratch paměti, kde je uložena hodnota globální proměnné
 - **globální proměnné programu 16-bitové** (`int`) přidáním prefixu „_“ a postfixů „_high“ a „_low“ pro přístup k vyššímu resp. nižšímu bytu proměnné. Zde

však pozor na to, že PCComp generuje uložení dvou bytů v paměti opačně. Např. `_xxx_low` je na adrese 3A a `_xxx_high` na 3B, zatímco při přidělování adres postupuje od konce paměti.

- **lokální parametry**, skrze vrcholu zásobníku umístěného ve scratch paměti
- **předat návratovou hodnotu funkce** skrze registry ZL, ZH pro nižší a vyšší byte hodnoty. PCComp na konec každé funkce přigeneruje vždy sekvenci příkazů `LOAD XL, ZL; LOAD XH, ZH; RETURN` (takže k předání se ve výsledku použijí registry XL a XH)
- **deklarace lokálních proměnných** musí být pouze na začátku funkcí a nesmí obsahovat inicializaci
- **deklarace funkcí** musí být přítomná pro všechny funkce kromě interrupt funkce, která je v assembleru, a samozřejmě deklarace musí předcházet prvnímu použití, takže je vhodné je umístit spolu s deklaracemi globálních proměnných na začátek souboru (implicitní deklarace funkce podle standardu jazyka C není zřejmě implementována)
- jediná optimalizace je v případě návratových příkazů z funkce, které jsou sdruženy do jediného RETURN-příkazu na konci funkce
- jediná další optimalizace, kterou lze aktivovat je vypuštění funkcí, které nejsou v programu volány (funkčnost zatím není dostatečně otestována)

Doporučení a upozornění

- pokuste se vyhnout jakémukoli kombinování více příkazů do jednoho nebo do jedné podmínky nebo do jednoho řádku; často se vám může stát, že kód bude bez varování chybně vygenerován!!!
- předávání proměnných do funkce odkazem není řádně otestováno a předpokládají se při tomto přístupu problémy
- pro cykly raději nepoužívejte `for` (až na případ `for(;;)`), protože zde například stejně nemůže být deklarovaná řídicí proměnná cyklu (někdy se stane, že skok `jump` na začátek cyklu je vygenerován chybně)
- násobení a dělení PCComp simuluje pomocí programu instrukcí, ale otázka jak efektivně (minimálně opět pravděpodobně používá 16-bitovou variantu i pro 8-bitové proměnné)
- funkce `interrupt` (v assembleru) je na konci obalena instrukcí `RETURNI DISABLE`, takže pokud chcete `RETURNI ENABLE` na konci ošetření přerušení, tak je potřeba na konec vložit tuto instrukci a tím udělat následující `RETURNI DISABLE` redundantní (nikdy nepřijde ke slovu, ale jak už bylo řečeno neobsahuje výsledný assembler PCComp žádné optimalizace a tudíž bude zbytečně zabírat další instrukci v instrukční paměti)

Překladač assembleru KCPSM3

- překladač assembleru si občas neporadí s mezerami navíc ve vloženém assembleru (především mezery za konstantami, které tvoří poslední argument instrukce)

Závěr

Dokument obsahuje naše aktuální poznatky o chybách a omezeních nalezených v současné verzi překladače C pro PicoBlaze od Francesca Poderica.

Architektura PicoBlaze je extrémní architektura z pohledu dostupných zdrojů paměti, instrukční paměti a nabízených operací. Použití procesoru PicoBlaze je zamýšleno pro specifickou třídu problémů (např. různá stavová řízení), zde se však vyžaduje maximálně optimalizovaný a funkční kód.

Velká chybovost překladače C pro PicoBlaze od Francesca Poderica znemožňuje reálné použití jazyka C při vývoji pro tuto architekturu. To vidíme jako obrovský motivační faktor hovořící jednoznačně pro realizaci kvalitního překladače C pro PicoBlaze.

Rozsah podpory standardu ANSI C je na zvážení, pro základní verzi PicoBlaze bez dodatečných paměťových IP komponent se zdá zbytečné implementovat standard kompletní – např. streamy a operace formátování výstupu svou rozsáhlostí překračují základní paměťové možnosti PicoBlaze (lze implementovat zpočátku rozsah nezbytný a po dodání externích paměťových IP rozšířit v dodávaných knihovnách).

Důraz by měl být kladen na správnou optimalizaci generovaného kódu (na počet instrukcí, na velikost použité operační paměti). Generování kódu může zohledňovat použití zásobníku v scratchpad paměti, na externích portech (tj. Port = paměťová buňka), či v externí paměti (bude-li dodán modul externí paměti připojitelný skrze porty).