

ADVANCED SIMULATOR OF MSP430 MICROCONTROLLERS

Jan Kaluža

Master Degree Programme (2), FIT BUT

E-mail: xkaluz09@stud.fit.vutbr.cz

Supervised by: Zdeněk Vašíček

E-mail: vasicek@fit.vutbr.cz

Abstract: This paper deals with the design and implementation of an advanced simulator of MSP430 microcontrollers. It describes the benefits of simulation and the basics of event driven simulation. The paper continues with a description of the design of the implemented instruction set simulator, MSP430 simulation library and graphical user interface used to control the simulation.

Keywords: simulation, MSP430, DEVS, microcontrollers

1 ÚVOD

Mikrokontrolery (MCU) jsou monolitické integrované obvody obsahující mikroprocesor, paměť pro uložení programu a jeho běh a další periferie rozšiřující jeho funkčnost. Vyznačují se jednoduchostí, kompaktností a malou spotřebou. Používají se tak proto především pro jednoúčelová zařízení a vestavěné (embedded) systémy.

Vzhledem k obtížnému způsobu vývoje a ladění vestavěných systémů vyžadujících často zapojení analyzátoru nebo osciloskopu je v praxi často používána simulace. Díky využití simulátoru daného mikrokontroleru (případně celého vestavěného systému) lze například ověřit funkčnost systému již při jeho návrhu, jednodušeji ladit chyby nebo zkoumat nové postupy bez dalšího zvýšení nákladů.

2 UDÁLOSTNĚ ŘÍZENÁ SIMULACE

Událostně řízená simulace [1] modeluje simulovaný systém jako sled diskrétních událostí. Každá událost se odehrává v přesně stanoveném čase a mění celkový stav systému. Mezi jednotlivými událostmi se stav systému nemění a je tak možné zpracovávat po sobě jdoucí události bez ohledu na časový interval mezi nimi.

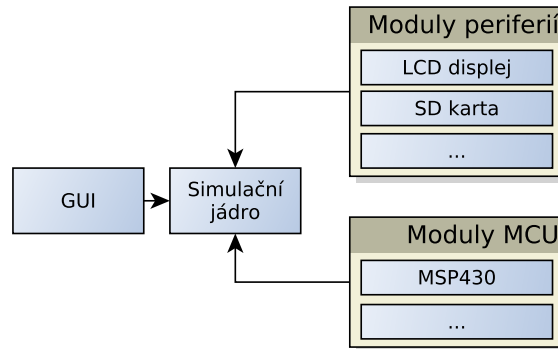
Samotná simulace funguje tak, že simulátor změní simulační čas na hodnotu první naplánované události v seznamu událostí. Tuto událost vykoná a pokračuje další naplánovanou událostí. Každá událost může při svém vykonávání přidat do seznamu událostí další položky. Simulace končí při dosažení ukončující podmínky (například omezení simulačním časem nebo při prázdném seznamu událostí).

3 POKROČILÝ MODULÁRNÍ SIMULÁTOR

Cílem mé práce bylo vytvoření multiplatformního modulárního simulátoru založeného na událostně řízené simulaci (konkrétně na formalismu DEVS [1]) schopného simulovat mikrokontrolery rodiny MSP430 [2] a další rozšiřující periferie, jako například LCD display nebo SD karta, komunikující s tímto mikrokontrolerem. V aktuální době nexistuje obdobný simulátor vydaný pod svobodnou licenci.

Simulátor se skládá ze 4 navzájem propojených částí znázorněných na obrázku 1. Základem je simulační jádro zajišťující řízení simulace, vzájemné propojení simulovaných periférií a jejich komunikaci. Díky modulárnímu návrhu jsou jednotlivé periferie a dokonce i mikrokontroler MSP430

implementovány jako přídavné moduly programované v jazyce C++ (v případě periférií pak i v jazyce Python). Lze tak jednoduše přidávat nové periférie nebo i zcela nové mikrokontrolery a tím simulátor rozšiřovat bez nutnosti zásahu do jeho interních částí. Uživatel simulátor ovládá pomocí uživatelského rozhraní, které je detailněji rozebráno v poslední kapitole tohoto příspěvku.



Obrázek 1: Struktura simulátoru.

Všechny periférie i mikrokontroler sdílí jednotné rozhraní pro komunikaci. Je jim umožněno vygenerovat na jednom ze svých pinů napětí (reprezentované pomocí desetinného čísla), které je pak v závislosti na propojení periférií předáno jako vstup jiné periférii. V rámci mé práce byly vytvořeny periférie LCD displej, SD karta, LED dioda, oscilátor a tlačítko.

4 SIMULACE MIKROKONTROLERU MSP430

Pro simulaci mikrokontroleru MSP430 byla vytvořena knihovna simulující funkci tohoto mikrokontroleru. Umožňuje nahrání uživatelského programu a jeho následnou simulaci pomocí vykonávání instrukcí a běhu svých dalších interních komponent, jako jsou například časovač, správce přerušení, modul BasicClock nebo moduly pro SPI komunikaci USART, USCI a USI.

Tato simulace je řízena simulačním jádrem. Knihovna také poskytuje grafickému uživatelskému rozhraní přístup ke své paměti a registrům, čímž umožňuje krokování programů a nabízí přístup k dalším ladícím informacím jako je například umístění lokálních proměnných v paměti nebo v registrech s využitím DWARF [3] ladících informací.

Jelikož je rozhraní knihovny obecné a není přímo svázáno s implementovaným simulátorem, je možné ji v budoucnu využít i k jiným projektům.

5 GRAFICKÉ UŽIVATELSKÉ ROZHRAŇÍ

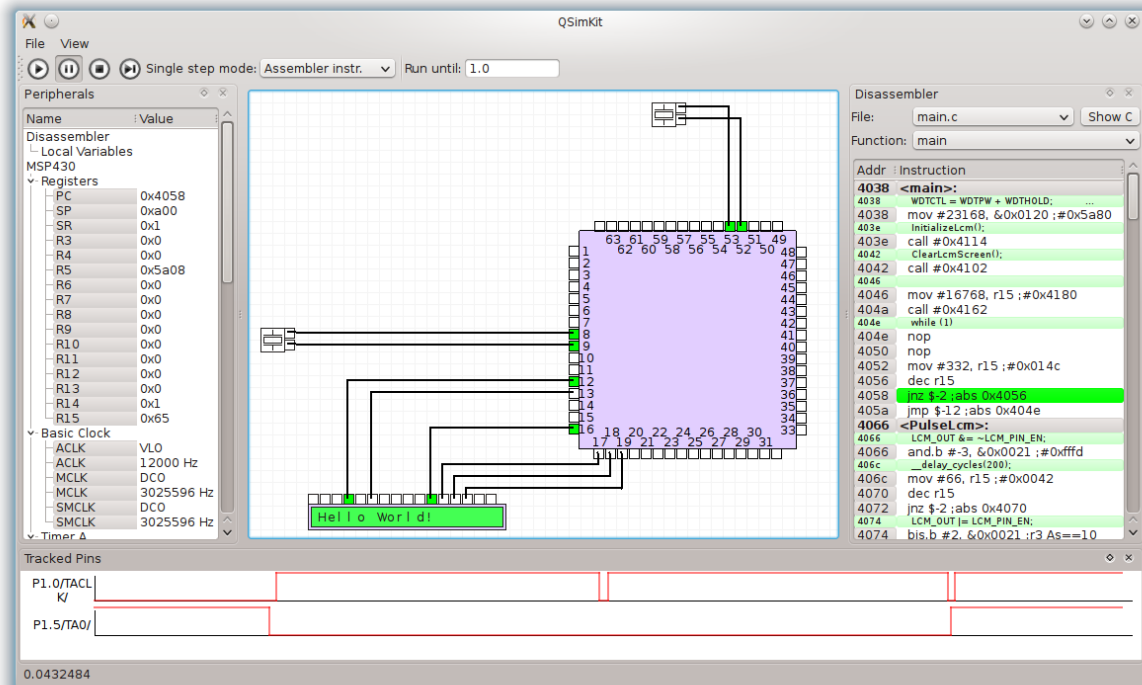
Grafické uživatelské rozhraní je zobrazeno na obrázku 2. Základním prvkem je kreslící plocha, na které jsou vykresleny jednotlivé simulované komponenty včetně pinů a propojení mezi piny. Uživatel je schopen přidávat nové komponenty a odstraňovat stávající, propojovat je pomocí vodičů a měnit jejich vlastnosti. Během simulace se komponenty aktualizují a uživatel tak vidí průběh simulace.

Další důležitou částí grafického uživatelského rozhraní je zobrazení zdrojového kódu programu nahráného v mikrokontroleru. Z principu lze vždy zobrazit kód v assembleru. Pokud je však dostupný zdrojový kód v jazyce C, je simulátor schopen přepínat mezi kódem v assembleru a kódem v C. Po zastavení simulace se zobrazí aktuální instrukce (případně odpovídající příkaz v jazyce C). Uživatelské rozhraní rovněž umožňuje nastavení breakpointů (bodů zastavení) na jednotlivých instrukcích.

Pro zobrazení interních informací o jednotlivých simulovaných komponentách slouží třetí část uživatelského rozhraní. Jde o strukturovaný seznam typu název-hodnota, který zobrazuje informace o ak-

tuálních hodnotách registrů mikroprocesoru, důležitých místech v paměti, nebo například hodnotách lokálních proměnných v místě zastavení simulace.

Poslední dílčí částí uživatelského rozhraní je rozhraní pro zobrazení napětí na jednotlivých pinech formou osciloskopu. Uživatel může vybrat piny, které chce během simulace sledovat a během simulace (nebo i po jejím zastavení) vidí veškeré změny signálu, které se na pinech odehrály.



Obrázek 2: Grafické uživatelské rozhraní.

6 ZÁVĚR

Tento příspěvek představil pokročilý, jednoduše rozšiřitelný, modulární simulátor, který může být v praxi využit pro zefektivnění vývoje a testování vestavěných systémů. Jeho grafické uživatelské rozhraní umožňuje pohodlné ladění programů řídicích vestavěných systém a jednoduché testování případných změn systému bez nutnosti využití reálného hardwaru. Simulace je přesná na jednotlivé instrukce (instruction accurate) a na běžném notebooku zpracuje až 350000 simulačních událostí za sekundu. Je zde však i prostor pro budoucí optimalizaci a výkon simulátoru tak může dále vzrůst. Další informace o tomto projektu lze nalézt na jeho oficiálních webových stránkách - <http://qsimkit.org>.

REFERENCE

- [1] Nutaro, J.: Building Software for Simulation: Theory and Algorithms, with Applications in C++. Hoboken, New Jersey: John Wiley and Sons, 2011, ISBN 978-0-470-41469-9
- [2] Davies, J.: MSP430 Microcontroller Basics. Burlington: Elsevier, 2008, ISBN 978-0-7506-8276-3, 686 s.
- [3] DWARF Standards Committee: DWARF Debugging Information Format. 1993, [Online], Poslední modifikace: 27. 7. 1993, [cit. 2014-02-28]. URL <http://www.dwarfstd.org/doc/dwarf-2.0.0.pdf>