

# FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



## Mikroprocesorové a vestavěné systémy – IMP

Měření výpočetních režii modulem  
Data Watchpoint and Trace (DWT)

## Obsah

1	Úvod.....	3
2	Vybavenie .....	3
3	Funkcionalita.....	3
3.1	Registre.....	3
3.1.1	DWT_CTRL.....	3
3.1.2	Počítacie registre.....	3
4	Riešenie.....	4
4.1	Globálne premenné .....	4
4.2	Inicializácia MCU.....	4
4.3	Inicializácia DWT.....	4
4.4	Počítanie réžie.....	4
4.4.1	Počítanie .....	4
4.4.2	Presnosť.....	5
5	Zhrnutie.....	5
6	Prílohy.....	5
7	Zdroje.....	6

# 1 Úvod

Úlohou bolo vytvoriť projekt demonštrujúci meranie výpočetnej réžie pomocou modulu Data Watchpoint and Trace (DWT) na mikrokontroléry Kinetis K60 s jadrom ARM Cortex-M4.

## 2 Vybavenie

Na použitie aplikácie je potreba FitKit3, na ktorom bude aplikácia bežať, počítač s aplikáciou Kinetis Design Studio (KDS) a Putty pre komunikáciu so zariadením.

Bohužiaľ v dobe pandémie FitKit3 nieje prístupný, takže odskúšanie aplikácie nebolo možné.

## 3 Funkcionalita

Aplikácia meria počet cyklov CPU, cyklov CPU strávených v spánku a cyklov CPU strávených spracovaním prerušení. Z výsledkov je zhrnutý celkový počet cyklov, maximum, minimum a priemer.

### 3.1 Registre

Stavové informácie o DWT poskytuje register DWT\_CTRL a používa sa na ovládanie funkcií DWT. Register DWT\_CTRL je zobrazený v obrázku č. 1. Na konkrétne počítanie cyklov sa využívajú počítacie registre

#### 3.1.1 DWT\_CTRL

Z DWT\_CTRL používame bity 0,18,19,20,21,24,25.

#### 3.1.2 Počítacie registre

Na počítanie používame 1 register na všetky cykly a na konkrétne merania profilové registre, ktorých DWT poskytuje 5. DWT\_CYCCNT na počítanie cyklov procesoru, DWT\_SLEEPCNT na počítanie cyklov počas, ktorých procesor spí, DWT\_EXCCNT na počítanie cyklov v prerušení, DWT\_LSUCNT na cykly vykonávajúce načítanie a uloženie(load & store), DWT\_FOLDCNT pre inštrukcie, ktoré zaberú 0 cyklov a DWT\_CPICNT počítajúci všetky inštrukcie . Príklady týchto registrov nájdeme na obrázkoch č. 2,3,4. Register DWT\_CYCCNT má 32 bitov. Profilové registre majú 8 bitov.

## 4 Riešenie

Celá implementácia sa nachádza v `main.c`. Implementácia obsahuje globálne premenné, základnú inicializáciu mikrokontroléru, inicializáciu modulu DWT a samotné počítanie réžie.

### 4.1 Globálne premenné

Máme zadefinované registre na ich patričných adresách.

DWT_CTRL	0xE0001000
DWT_CYCCNT	0xE0001004
DWT_EXCCNT	0xE000100C
DWT_CPICNT	0xE0001008
DWT_SLEEPCNT	0xE0001010
DWT_LSUCNT	0xE0001014
DWT_FOLDCNT	0xE0001018

Premenné kde sa uloží začiatok a koniec počítania a premenné pre počet cyklov, maximum, minimum a priemer. Nachádza sa tu aj premenná, ktorá určuje počet opakovaní meraného kódu.

### 4.2 Inicializácia MCU

Základné nastavenie hodín a vypnutie watchdogu.

### 4.3 Inicializácia DWT

Zapnutie DWT\_CTRL nastavením nultého bitu na 1. Skontrolovanie bitu 24, ktorý určuje či implementácia podporuje čítače na výnimky, spánok, LSU a čítač inštrukcií, ktoré zaberú 0 cyklov. Ďalej bit 25, ktorý určuje či implementácia podporuje čítač cyklov procesora. V oboch prípadoch sa bit musí rovnať 0 aby indikoval, že sú podporované.

### 4.4 Počítanie réžie

#### 4.4.1 Počítanie

Čítače sa vynulujú a ich hodnoty sa uložia do premenných. DWT\_CYCCNT sa priradí 0 a to ho resetuje. DWT\_LSUCNT, DWT\_FOLDCNT, DWT\_SLEEPCNT, DWT\_EXCCNT a DWT\_CPICNT sa vynulujú bitmi 21, 20, 19, 18 a 17 v registri DWT\_CTRL, v cykle sa vykoná kód určený na meranie, finálne hodnoty čítačov sa uložia do premenných, vypočítajú sa požadované maximá, minimá a priemery a vypíšu sa do terminálu. Premenné sa vynulujú a môže sa merať ďalší kus kódu určený na meranie réžie.

Profilové registre po napočítaní 256 cyklov pretečú a nastaví príslušný bit na 1.

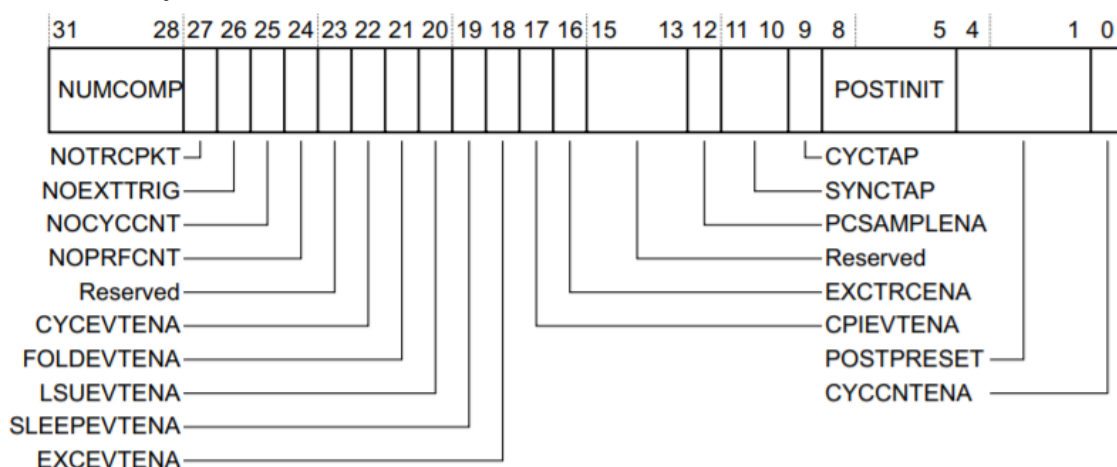
#### 4.4.2 Presnosť

Pri veľkých meraniach môže DWT\_FOLDCNT silne ovplyvniť presnosť výpočtov. Menšie nepresnosti sú však povolené, nie je však jasne určené aká veľká nepresnosť je už nežiadúca.

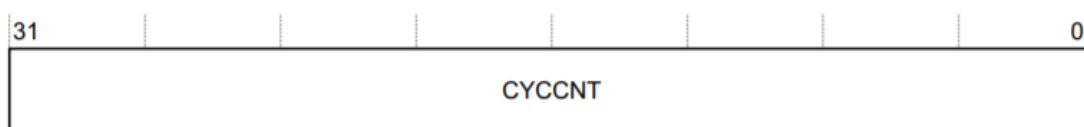
### 5 Zhrnutie

Projekt som bez FitKitu3 nemohol vyskúšať, ale snažil som sa čo najvernejšie priblížiť funkčnému riešeniu. Podľa viacerých zdrojov a hlavne manuálu ARM®v7-M Architecture Reference Manual, ktorý poskytol všetky potrebné informácie k práci s DWT som bol schopný projekt vytvoriť. Problém v implementácii by som videl v spôsobe spracovania a konkrétne v prípade merania kedy sa vykonávajú všetky merania naraz, a testované kódy sa volajú cez funkcie, čo môže viesť k skresleniu výpočtov. Bez možnosti odskúšať projekt je implementácia riešená takto kvôli prehľadnosti. Nepodarilo sa mi implementovať meranie riadiacich algoritmov s prerušením a bez.

### 6 Prílohy



Obr. 1 – register DWT\_CTRL



Obr. 2 – register DWT\_CYCCNT



Obr. 3 – register DWT\_EXCCNT



Obr. 4 – register DWT\_SLEEPCNT

## 7 Źródło

1. ARM® Cortex®-M4 Processor Technical Reference Manual
2. K60 Sub-Family Reference Manual,  
Document Number: K60P144M100SF2V2RM Rev. 2 Jun 2012
3. ARM®v7-M Architecture Reference Manual
4. <https://www.embedded-computing.com/articles/measuring-code-execution-time-on-arm-cortex-m-mcus>
5. <https://arm-stm.blogspot.com/2014/05/dwt-data-watchpoint-and-trace-unit.html>