

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Simulátor mikroprocesoru AVR



2013

Martin Heitel

Anotace

Simulátor AVR demonstruje činnost mikroprocesoru AVR. Uživatel má možnost sledovat vykonávání vloženého programu po krocích, sledovat obsah obecných registrů, obsah speciálních registrů, obsah paměti RAM a úroveň signálů na portech. Pro lepší názornost je úroveň signálů na portech zobrazována také na připojených perifériích v podobě pole LED a displeje. Nabízí uživateli uživatelsky jednoduchý a přívětivý nástroj pro testování kódů v AVR Assembleru pro mikroprocesory AVR.

Děkuji vedoucímu práce RNDr. Arnoštu Večerkovi za odbornou pomoc při tvorbě mé bakalářské práce.

Obsah

1. Úvod	6
1.1. Existující řešení	6
1.1.1. VMLAB	6
1.1.2. WinAVR	6
1.2. Návrh řešení	7
2. Popis mikroprocesoru AVR	8
2.1. Mikroprocesor AVR	8
2.1.1. Obecné registry	8
2.1.2. Speciální registry	8
2.1.3. Paměť RAM	8
2.1.4. Programová paměť	8
2.1.5. Porty	8
2.1.6. Periferie	8
2.1.7. AVR Assembler	9
2.1.8. Instrukční sada	9
2.1.9. Řady mikroprocesorů AVR	9
2.2. Rozdíly oproti Intel 8051	9
3. Uživatelská dokumentace	10
3.1. Instalace	10
3.2. Spuštění	10
3.3. Hlavní okno	10
3.4. Menu programu	11
3.4.1. Menu	11
3.4.2. Panel nástrojů	11
3.5. Okno Pole LED	12
3.6. Okno Displej	12
3.7. Okno Klávesnice	12
3.8. Okno Nastavení	12
3.9. Okno Náповěda	12
3.10. Okno AVR Assembler	12
3.11. Okno O Aplikaci	12
4. Programátorská dokumentace	13
4.1. Použité technologie a nástroje	13
4.2. Použité třídy a metody	13
4.2.1. Třída Sreg	13
4.2.2. Třída Regs	13
4.2.3. Třída Ram	14
4.2.4. Třída PrevodySoustav	14

4.2.5.	Třída Port	15
4.2.6.	Třída MPC	15
4.2.7.	Třída LEDpole	17
4.2.8.	Třída Dioda	17
4.2.9.	Třída FormNastaveni	17
4.2.10.	Třída FormNapoveda	17
4.2.11.	Třída FormAVRassemblerNapoveda	18
4.2.12.	Třída Displej	18
4.2.13.	Třída AboutBox1	19
4.2.14.	Třída Form1	19
4.3.	Windows Forms	26
4.3.1.	Form1	26
4.3.2.	LEDpole	28
4.3.3.	Displej	28
4.3.4.	Klavesnice	28
4.3.5.	FormNastaveni	28
4.3.6.	FormNapoveda	28
4.3.7.	FormAVRassemblerNapoveda	29
4.3.8.	AboutBox1	29
4.4.	Sestavení aplikace	29
	Závěr	30
	Reference	31
	A. Obsah přiloženého CD	32

1. Úvod

Tento Simulátor AVR jsem vytvořil pro potřeby demonstrace činnosti mikroprocesoru AVR. Simuluje činnost mikroprocesorů typické řady ATmega. Umožňuje uživateli vložit do Simulátoru AVR program - zdrojový kód v AVR Assembleru a sledovat vykonávání vloženého programu po jednotlivých krocích. V jednotlivých krocích umožňuje sledovat aktuálně vykonávanou instrukci a instrukce v jejím okolí, obsah obecných registrů, obsah speciálních registrů, obsah paměti RAM a dále úroveň signálů na portech. Pro lepší názornost je možno připojit periferie v podobě pole LED a displeje, na kterých je úroveň signálů na portech také zobrazována.

Simulátor AVR je řešen s důrazem na jednoduché, přívětivé uživatelské prostředí a je určen zejména pro začátečníky v oboru programování AVR mikroprocesorů pro účel kontroly zdrojových kódů.

1.1. Existující řešení

Před započítím prací na tomto projektu jsem provedl průzkum existujících řešení. Výsledkem průzkumu bylo zjištění, že existující řešení poskytují široké možnosti, většinou však neposkytují možnost jednoduché obsluhy pro začátečníka v oboru tvorby kódů programu pro mikroprocesory AVR.

Nejvíce inspirativní byly simulátory (vývojová prostředí) VMLAB a WinAVR.

1.1.1. VMLAB

Poskytuje možnosti vytvořit kompletní virtuální prototyp aplikace pro mikroprocesory AVR. Tento simulátor jsem ohodnotil jako uživatel nejlépe, i když pro začátečníka se může jevit komplikovaně.

1.1.2. WinAVR

Sdružuje open source nástroje pro vývoj aplikací pro mikroprocesory AVR. Uživatelské rozhraní se jeví jednodušeji než u VMLAB, ale seznámení se s programem a spuštění první simulace mi u WinAVR trvalo déle než u VMLAB.

1.2. Návrh řešení

Již při průzkumu existujících řešení jsem hledal simulátor, který by byl jednoduchý na obsluhu. Jednoduchost obsluhy simulátoru a možnost rychle spustit kód programu pro mikroprocesor AVR se tak stal jedním z hlavních cílů pro mé řešení. Dalšími pak jsou přehledné zobrazení důležitých informací při běhu programu v jednom okně (obsah registrů, obsah paměti RAM, obsah paměti pro program, aktuálně vykonávaná instrukce, úroveň signálů na portech).

2. Popis mikroprocesoru AVR

V následující části stručně popíši součásti mikroprocesoru AVR, které jsou zobrazovány Simulátorem AVR a některé rozdíly oproti Intel 8051. Zájemce o podrobnější popis odkazují na [1], [2], [3].

2.1. Mikroprocesor AVR

Stručný popis mikroprocesoru AVR :

2.1.1. Obecné registry

Obecné registry se skládají z 32 stejných 8-bitových registrů. V těchto registrech mohou být uložena jak data, tak adresy. Posledních 6 registrů můžeme ve dvojicích využívat jako X, Y, Z registry sloužící jako ukazatele adresy pro účely nepřímého adresování paměti.

2.1.2. Speciální registry

Neboli Stavový registr je registr v I/O prostoru. Jeho jednotlivé bity jsou příznaky přenosu (C), nulového výsledku (Z), záporného výsledku (N), přetečení dvojkového doplňku (V), znaménko výsledku (S), přenosu mezi třetím a čtvrtým bitem (H), transfer bit (T) a povolení přerušení (I).

2.1.3. Paměť RAM

Je paměť pro data. Typická velikost je 512 x 8 bitů.

2.1.4. Programová paměť

Typická velikost je 4096 x 16 bitů, paměť typu Flash.

2.1.5. Porty

Mohou sloužit pro komunikaci s okolím. Mohou k nim být připojeny periferie. Různé řady mikroprocesorů mají různý počet portů, typické jsou 4.

2.1.6. Periferie

Periferií může být například pole LED, segmentový displej, klávesnice a jiné.

2.1.7. AVR Assembler

Nízkoúrovňový programovací jazyk tvořený symbolickou reprezentací strojových instrukcí a konstant. Je využíván pro vytvoření strojového kódu programu pro procesor.

2.1.8. Instrukční sada

Soubor strojových instrukcí (RISC - redukováná instrukční sada) implementovaných přímo v mikroprocesoru.

2.1.9. Řady mikroprocesorů AVR

Jednotlivé řady mikroprocesorů se liší zejména svou instrukční sadou. Dále také počtem vstupně/výstupních portů, velikostí obou typů pamětí a dalšími rozdíly. Pobornější výčet rozdílů naleznete v [1] a [4].

Nejznámější řady mikroprocesorů AVR jsou :

- ATtiny
- ATmega
- ATxmega
- Automotive AVR
- 32-bit AVR

2.2. Rozdíly oproti Intel 8051

- Instrukční slovo 16 bitů.
- Propojení ALU (aritmeticko-logická jednotka) s pracovními (obecnými) registry.
- Až 12x větší výpočetní výkon při stejné frekvenci.
- Nastavení příznaku Z u některých instrukcí (například CPC, SBC, SBCI).
- Rozsahy použitelných registrů jako operandů u instrukcí.

3. Uživatelská dokumentace

Simulátor AVR je aplikace pro PC s operačním systémem Windows (XP / Vista / 7).

3.1. Instalace

K dispozici je složka s instalačními soubory. Instalace se spouští spuštěním souboru Setup.exe. V případě, že na PC, na který chceme instalovat aplikaci, není nainstalována odpovídající verze NET Framework, instalační soubor nabídne instalaci odpovídající verze NET Framework on-line (vyžadováno připojení k internetu).

3.2. Spuštění

Zástupce pro spuštění Simulátoru AVR je umístěn v nabídce Start.

3.3. Hlavní okno

Hlavní okno Simulátoru AVR umožňuje sledovat :

- Obsah 32 pracovních registrů (R0 až R31).
- Obsah 3 zdvojených registrů X, Y, Z.
- Obsah speciálních registrů - FLAGS I, T, H, S, V, N, Z, C.
- Obsah vstupně/výstupních portů A, B, C, D.
- Kód programu vykonávaného mikroprocesorem v AVR Assembleru. (Aktuálně vykonávaná instrukce je označena).
- Obsah paměti mikroprocesoru pro program v jazyce mikroprocesoru. (Aktuálně vykonávaná instrukce je označena).
- Obsah paměti mikroprocesoru pro data. (V jazyce mikroprocesoru).

3.4. Menu programu

Ovládací prvky Simulátoru AVR jsou soustředěny v Menu a Panelu nástrojů.

3.4.1. Menu

- Soubor :
 - Nový - Zresetování simulátoru.
 - Otevřít - Výběr a načtení souboru, ve kterém je uložen program v AVR Assembleru.
 - Exit - Ukončení Simulátoru.
- Úpravy :
 - Stop - Pozastavení běhu programu.
 - Pokračovat - Pokračování běhu programu.
 - Další krok - Vykonání následující instrukce.
 - Periferie :
 - Pole LED - Aktivuje okno s polem LED diod.
 - Displej - Aktivuje okno s numerickým displejem.
 - Klávesnice - Aktivuje okno s numerickou klávesnicí.
 - Všechny periferie - Aktivuje Pole LED, Displej i Klávesnici.
 - Nastavení - Nastavení délky časového intervalu mezi vykonávanými instrukcemi.
- Nápověda :
 - Nápověda - Popis a nápověda k ovládání Simulátoru.
 - AVR Assembler - Seznam instrukcí AVR Assembleru.
 - O aplikaci - Informace o Simulátoru (Verze / Licence / Autor).

3.4.2. Panel nástrojů

- Nový - Zresetování Simulátoru.
- Otevřít - Výběr a načtení souboru, ve kterém je uložen program v AVR Assembleru.
- Stop - Pozastavení běhu programu.
- Pokračovat - Pokračování běhu programu.

- Další krok - Vykonání následující instrukce.
- Pole LED - Aktivuje okno s polem LED diod.
- Displej - Aktivuje okno s numerickým displejem.
- Klávesnice - Aktivuje okno s numerickou klávesnicí.
- Nastavení - Nastavení délky časového intervalu mezi vykonávanými instrukcemi.
- AVR Assembler - Seznam instrukcí AVR Assembleru.

3.5. Okno Pole LED

V okně se zobrazí pole LED a rozsvícené / nerozsvícené diody reprezentují hodnoty na portech. Slouží jako výstupní periferie.

3.6. Okno Displej

V okně se zobrazí displej a zobrazená čísla reprezentují hodnoty na portech. Slouží jako výstupní periferie.

3.7. Okno Klávesnice

V okně se zobrazí klávesnice, kterou je možno připojit k jednomu z portů. Slouží jako vstupní periferie.

3.8. Okno Nastavení

Umožňuje nastavit časový interval mezi vykonávanými instrukcemi.

3.9. Okno Nápověda

V okně se zobrazí nápověda k Simulátoru AVR.

3.10. Okno AVR Assembler

V okně se zobrazí nápověda - seznam instrukcí AVR Assembleru.

3.11. Okno O Aplikaci

V okně se zobrazí informace k Simulátoru AVR (Verze / Licence / Autor).

4. Programátorská dokumentace

4.1. Použité technologie a nástroje

Aplikace je vytvořena v prostředí MS Visual C# 2008 a tudíž pro svůj běh potřebuje .NET Framework 3.5.

4.2. Použité třídy a metody

Použité třídy a metody :

4.2.1. Třída Sreg

Třída a metody pro práci se Speciálními registry - Flags.

Obsahuje následující metody :

- public Sreg() - konstruktor.
- public void SetSREG(char reg, byte stav) - nastavení hodnoty bitu v SREG.
- public byte GetSREG(char reg) - vrátí hodnotu bitu v SREG.
- public void ClearAllSREG() - nastavení hodnoty 0 ve všech bitech SREG.
- public string GetAllSREG() - vrátí hodnoty všech bitů v SREG.
- public void SetAllSREG(string bin) - nastavení hodnoty všech bitů v SREG.

4.2.2. Třída Regs

Třída a metody pro práci s pracovními registry.

Obsahuje následující metody :

- public Regs() - konstruktor.
- public void SetReg(byte reg, string hodnota) - nastavení hodnoty registru.
- public string GetReg(byte reg) - vrátí hodnotu registru.
- public void ClearAllReg() - vymaže hodnoty v pracovních registrech.

4.2.3. Třída Ram

Třída a metody pro práci s pamětí.

Obsahuje následující metody :

- public Ram() - konstruktor.
- public void DataClear() - vymaže hodnoty v paměti RAM.
- public void ProgramClear() - vymaže hodnoty v paměti Flash.
- public void RamClear() - vymaže hodnoty v paměti Flash i RAM.
- public int ProgramMAX() - vrátí velikost paměti Flash.

4.2.4. Třída PrevodySoustav

Třída a metody pro práci s převody čísel do různých číselných soustav.

Obsahuje následující metody :

- public static string DecToBin(int cislo)
Vrací číslo v binární soustavě, vstupem je dekadická hodnota.
- public static string DecToBin5(int cislo)
Vrací číslo v binární soustavě o délce 5 znaků, vstupem je dekadická hodnota.
- public static string DecToBin22(int cislo)
Vrací číslo v binární soustavě o délce 22 znaků, vstupem je dekadická hodnota.
- public static int BinToDec(string bincislo)
Vrací číslo v dekadické soustavě, vstupem je binární hodnota.
- public static string DvojDoplnek(string bin)
Vrací dvojkový doplněk, vstup i výstup ve dvojkové soustavě.
- public static string DecToHex(int cislo)
Vrací číslo v hexadecimální soustavě, vstupem je dekadická hodnota.
- public static int HexToDec(string hexcislo)
Vrací číslo v dekadické soustavě, vstupem je hexadecimální hodnota.
- public static string BinToHex(string bin)
Vrací číslo v hexadecimální soustavě, vstupem je binární hodnota.

- public static string BinToHex4(string bin)
Vrací číslo v hexadecimální soustavě o délce 4 znaky, vstupem je binární hodnota.
- public static string HexToBin(string hex)
Vrací číslo v binární soustavě o délce 8 znaků, vstupem je hexadecimální hodnota.
- public static byte Prenos3bitADD(string Rd, string Rr, string R)
Vrací 0 nebo 1 v závislosti na přenosu mezi 3 a 4 bitem pro funkci sčítání.
- public static byte Prenos3bitSUB(string Rd, string Rr, string R)
Vrací 0 nebo 1 v závislosti na přenosu mezi 3 a 4 bitem pro funkci odečítání a porovnání.

4.2.5. Třída Port

Třída a metody pro práci s porty.

Obsahuje následující metody :

- public void SetPortDec(byte hodnota)
Nastaví hodnotu na portu, vstupem je dekadická hodnota.
- public void SetPortBin(string hodnota)
Nastaví hodnotu na portu, vstupem je binární hodnota.
- public string GetPortBin()
Vrátí binární hodnotu portu.
- public byte GetPortDec()
Vrátí dekadickou hodnotu portu.

4.2.6. Třída MPC

Třída a metody pro práci s mikroprocesorem.

Využívá třídy Sreg, Regs, Port a Ram.

Obsahuje následující metody :

- public MPC()
Konstruktor.
- public void SetSREG(char reg, byte stav)
Nastaví hodnotu SREG.

- public byte GetSREG(char reg)
Vrátí hodnotu SREG.
- public void SetReg(byte reg, string hodnota)
Nastaví hodnotu v pracovním registru.
- public void SetXYZhex(char xyz, string hex)
Nastaví hodnoty ve zdvojeném registru (X, Y nebo Z).
- public string GetReg(byte reg)
Vrátí hodnotu v pracovním registru.
- public void ClearMPC()
Reset, vymazání všech hodnot (Paměť, registry, porty, ...).
- public void SetEIND(string DecCislo)
Nastaví hodnotu v registru EIND.
- public string GetEINDhex()
Vrátí hodnotu v registru EIND.
- public void StackAddCounter()
Uloží aktuální hodnotu Counter na zásobník.
- public void StackTakeLastCounter()
Nastaví Counter na hodnotu uloženou v zásobníku.
- public void StackPushReg(int regIndex)
Uloží hodnotu uloženou v registru na zásobník.
- public void StackPopReg(int regIndex)
Uloží hodnotu uloženou v zásobníku do registru.
- public void SetPortBin(string port, string hodnota)
Uloží hodnotu na port, vstupem je binární hodnota.
- public void SetPortDec(string port, byte hodnoty)
Uloží hodnotu na port, vstupem je dekadická hodnota.
- public string GetPortBin(string port)
Vrátí hodnotu uloženou na portu v binární hodnotě.
- public byte GetPortDec(string port)
Vrátí hodnotu uloženou na portu v dekadické hodnotě.
- public void AktualizaceRegistru()
Pomocná metoda zajišťující korektnost obsahu paměti, portů a registrů.

4.2.7. Třída LEDpole

Třída a metody pro práci s LED polem.

Využívá třídu Dioda.

Obsahuje následující metody :

- public LEDpole()
Konstruktor.
- private void LEDpole_FormClosing(object sender, FormClosingEventArgs e)
Zavření okna.

4.2.8. Třída Dioda

Třída a metody pro práci Diodou.

Obsahuje následující metody :

- public Dioda(int x, int y, int sirka, int vyska)
Konstruktor.
- public void dioda_Paint(object sender, PaintEventArgs e)
Zobrazení diody.

4.2.9. Třída FormNastaveni

Třída a metody pro práci s oknem pro nastavení.

Obsahuje následující metody :

- public FormNastaveni(int t)
Konstruktor.
- private void button1_Click(object sender, EventArgs e)
Zavření okna.

4.2.10. Třída FormNapoveda

Třída a metody pro práci s oknem pro nápovědu.

Obsahuje následující metody :

- public FormNapoveda()
Konstruktor.

- private void timer1_Tick(object sender, EventArgs e)
Udržení textu nápovědy v okně.
- private void FormNapoveda_FormClosed(object sender, FormClosedEventArgs e)
Zavření okna.

4.2.11. Třída FormAVRassemblerNapoveda

Třída a metody pro práci s oknem pro nápovědu k AVR Assembleru.

Obsahuje následující metody :

- public FormAVRassemblerNapoveda()
Konstruktor.
- private void timer1_Tick(object sender, EventArgs e)
Udržení textu nápovědy v okně.
- private void FormAVRassemblerNapoveda_FormClosing(object sender, FormClosingEventArgs e)
Zavření okna.

4.2.12. Třída Displej

Třída a metody pro práci s oknem pro displej.

Obsahuje následující metody :

- public Displej()
Konstruktor.
- public PictureBox[] cisloDisleje(int x, int y, Color On, Color Off)
Umístění číslice displej v okně.
- public string CisloNaDisplej(string bin)
Vrátí hodnotu pro zobrazení na displeji.
- private void Displej_FormClosing(object sender, FormClosingEventArgs e)
Zavření okna.

4.2.13. Třída AboutBox1

Třída a metody pro práci s oknem pro zobrazení informací k programu (Autor, verze, licence).

Obsahuje následující metody :

- public AboutBox1()
Konstruktor.
- private void buttonOK_Click(object sender, EventArgs e)
Zavření okna.
- private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
Spuštění programu pro napsání e-mailu po kliknutí na e-mailovou adresu.

4.2.14. Třída Form1

Třída a metody pro práci s hlavním oknem Simulátoru AVR.

Obsahuje následující metody :

- public Form1()
Konstruktor.
- public void Zobrazit()
Metoda pro zobrazování požadovaných hodnot v okně.
- public void Chyba(string jaka)
Metoda pro zobrazování požadovaných hodnot v okně, pokud je chyba v souboru s kódem v AVR Assembleru.
- private void openToolStripMenuItem_Click(object sender, EventArgs e)
Otevření a načtení souboru s kódem v AVR Assembleru (povelem z nabídky).
- private void timer1_Tick(object sender, EventArgs e)
Vykonávání instrukcí načtených ze souboru s kódem v AVR Assembleru.
- private void exitToolStripMenuItem_Click(object sender, EventArgs e)
Zavření okna povelem z nabídky.
- private void newToolStripMenuItem_Click(object sender, EventArgs e)
Reset mikroprocesoru, vymazání všech uložených hodnot (povelem z nabídky).

- private void Form1_FormClosing(object sender, FormClosingEventArgs e)
Provedení zavření okna.
- public void nastaveníToolStripMenuItem_Click(object sender, EventArgs e)
Zobrazení okna pro nastavení (povelem z nabídky).
- private void pauzeToolStripMenuItem_Click(object sender, EventArgs e)
Pozastavení vykonávání instrukcí (povelem z nabídky).
- private void pokračovatToolStripMenuItem_Click(object sender, EventArgs e)
Pokračovat ve vykonávání instrukcí (povelem z nabídky).
- private void nextToolStripMenuItem_Click(object sender, EventArgs e)
Provést následující instrukci (povelem z nabídky).
- private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
Zobrazení okna s informacemi o programu AVR Simulátor povelem z nabídky (Autor, verze, licence).
- private void toolStripMenuItem1_Click(object sender, EventArgs e)
Zobrazení okna s nápovědou k programu AVR Simulátor (povelem z nabídky).
- private void contentsToolStripMenuItem_Click(object sender, EventArgs e)
Zobrazení okna s nápovědou k AVR assembleru (povelem z nabídky).
- private void poleLEDToolStripMenuItem_Click(object sender, EventArgs e)
Zobrazení okna s polem LED (povelem z nabídky).
- private void displejToolStripMenuItem_Click(object sender, EventArgs e)
Zobrazení okna s displejem (povelem z nabídky).
- private void klávesniceToolStripMenuItem_Click(object sender, EventArgs e)
Zobrazení okna s klávesnicí (povelem z nabídky).
- private void všechnyPeriferieToolStripMenuItem_Click(object sender, EventArgs e)
Zobrazení oken s klávesnicí, displejem a polem LED (povelem z nabídky).
- private void listBoxKodASM_SelectedIndexChanged(object sender, EventArgs e)
Při změně označeného řádku v listBoxKodASM se mění označený řádek také v listBoxProgram.

- private void listBoxProgram_SelectedIndexChanged(object sender, EventArgs e)
Při změně označeného řádku v listBoxProgram se mění označený řádek také v listBoxKodASM.
- private List<string> odsadit_navesti(List<string> kod)
Odsazení instrukcí a argumentů instrukcí pro lepší přehlednost v listBoxKodASM.
- private void newToolStripButton_Click(object sender, EventArgs e)
Reset mikroprocesoru, vymazání všech uložených hodnot (povelem z panelu nástrojů).
- private void openToolStripButton_Click(object sender, EventArgs e)
Otevření a načtení souboru s kódem v AVR Assembleru (povelem z panelu nástrojů).
- private void pauzeToolStripButton_Click(object sender, EventArgs e)
Pozastavení vykonávání instrukcí (povelem z panelu nástrojů).
- private void pokracovatToolStripButton_Click(object sender, EventArgs e)
Pokračovat ve vykonávání instrukcí (povelem z panelu nástrojů).
- private void nextToolStripButton_Click(object sender, EventArgs e)
Provést následující instrukci (povelem z panelu nástrojů).
- private void PoleLedtoolStripButton_Click(object sender, EventArgs e)
Zobrazení okna s polem LED (povelem z panelu nástrojů).
- private void displaytoolStripButton_Click(object sender, EventArgs e)
Zobrazení okna s displejem (povelem z panelu nástrojů).
- private void klavesnicetoolStripButton_Click(object sender, EventArgs e)
Zobrazení okna s klávesnicí (povelem z panelu nástrojů).
- private void helpToolStripButton_Click(object sender, EventArgs e)
Zobrazení okna s nápovědou k AVR assembleru (povelem z panelu nástrojů).
- private void nastaveniToolStripButton_Click(object sender, EventArgs e)
Zobrazení okna pro nastavení (povelem z panelu nástrojů).
- public int asm(string radek)
Analýza řádku z kódu v AVR Assembleru, zjištění instrukce a argumentů.
- public int NajdiCisloRadkuBezNavestiRozdil(string povel)
Vrátí rozdíl čísla řádku s instrukcí (řádky s návěstím se nezapočítávají) a čísla aktuálního řádku.

- public int NajdiCisloRadkuBezNavesti(string povel)
Vrátí číslo řádku s instrukcí (řádky s návěstím se nezapočítávají).
- public string zapisLDdoRAM(byte reg, string adresa)
Překlad instrukce LD() pro uložení do paměti.
- public string zapisLDDdoRAM(byte reg, string adresa)
Překlad instrukce LDD() pro uložení do paměti.
- public string zapisINdoRAM(byte reg, string odkud)
Překlad instrukce pro uložení do paměti.
- public string zapisOUTdoRAM(string kam, byte reg)
Překlad instrukce pro uložení do paměti.
- public string zapisSTdoRAM(string adresa, byte reg)
Překlad instrukce ST() pro uložení do paměti.
- public string zapisSTDdoRAM(string adresa, byte reg)
Překlad instrukce STD() pro uložení do paměti.
- Metody simulující činnost instrukcí AVR Assembleru. Tyto instrukce představují instrukční soubor mikroprocesoru AVR. Při tvorbě těchto metod jsem vycházel z popisu instrukční sady v [1], [2] a [3], kde jsou popsány jednotlivé instrukce. Nejpodrobnější popis je uveden v [1].
 - public void LDI(byte reg, int k)
 - public void LDS(byte reg, int k)
 - public void LD(byte reg, string adresa)
 - public void STS(int adresa, byte reg)
 - public void ST(string adresa, byte reg)
 - public void STD(string adresa, byte reg)
 - public void ADD(byte reg1, byte reg2)
 - public void ADC(byte reg1, byte reg2)
 - public void ADIW(int Rd, byte K)
 - public void AND(byte reg1, byte reg2)
 - public void TST(byte Rd)
 - public void OR(byte Rd, byte Rr)
 - public void ANDI(byte reg1, int K)
 - public void CBR(byte Rd, int K)
 - public void ORI(byte reg1, int K)

- public void SBR(byte Rd, int K)
- public void EOR(byte Rd, byte Rr)
- public void ASR(byte Rd)
- public void BCLR(byte s)
- public void BLD(byte reg, byte b)
- public void BRBC(byte s, string kam)
- public void BRBS(byte s, string kam)
- public void BREQ(string kam)
- public void BRGE(string kam)
- public void BRHC(string kam)
- public void BRHS(string kam)
- public void BRID(string kam)
- public void BRIE(string kam)
- public void BRLO(string kam)
- public void BRLT(string kam)
- public void BRMI(string kam)
- public void BRNE(string kam)
- public void BRPL(string kam)
- public void BRSH(string kam)
- public void BRTC(string kam)
- public void BRTS(string kam)
- public void BRVC(string kam)
- public void BRVS(string kam)
- public void BSET(byte s)
- public void BST(byte reg, byte b)
- public void MOV(byte Rd, byte Rr)
- public void MOVW(int Rd, int Rr)
- public void MUL(byte Rd, byte Rr)
- public void MULS(byte Rd, byte Rr)
- public void MULSU(byte Rd, byte Rr)
- public void FMUL(byte Rd, byte Rr)
- public void FMULS(byte Rd, byte Rr)

- public void FMULSU(byte Rd, byte Rr)
- public void SUB(byte Rd, byte Rr)
- public void SUBI(byte Rd, int k)
- public void SBCI(byte Rd, int k)
- public void SBIW(byte Rd, int K)
- public void SBC(byte Rd, byte Rr)
- public void NOP()
- public void SEZ()
- public void SEV()
- public void SET()
- public void SES()
- public void SEN()
- public void SEC()
- public void SEI()
- public void SEH()
- public void CLC()
- public void CLH()
- public void CLI()
- public void CLN()
- public void CLS()
- public void CLT()
- public void CLV()
- public void CLZ()
- public void CLR(byte Rd)
- public void SER(byte Rd)
- public bool IJMP()
- public bool EIJMP()
- public void OUT(string kam, byte Rr)
- public void IN(byte reg, string odkud)
- public void SBI(int IOprostor, int b)
- public void CBI(int IOprostor, int b)
- public void LSL(byte regIndex)

- public void ROL(byte regIndex)
- public void LSR(byte regIndex)
- public void ROR(byte regIndex)
- public void SWAP(byte regIndex)
- public void PUSH(int regIndex)
- public void POP(int regIndex)
- public void CALL(string kam)
- public void RCALL(string kam)
- public bool ICALL()
- public bool EICALL()
- public void RET()
- public void RETI()
- public void BRCC(string kam)
- public void BRCS(string kam)
- public void DEC(byte Rd)
- public void INC(byte Rd)
- public void NEG(byte Rd)
- public void COM(byte Rd)
- public void CPSE(byte Rd, byte Rr)
- public void SBRC(byte Rr, byte b)
- public void SBRS(byte Rr, byte b)
- public void SBIC(int IOint, byte b)
- public void SBIS(int IOint, byte b)
- public void CP(byte Rd, byte Rr)
- public void CPC(byte Rd, byte Rr)
- public void CPI(byte Rd, int K)
- public void LPM(byte reg, string regZ)

4.3. Windows Forms

Simulátor využívá následující Windows Forms:

4.3.1. Form1

V této Windows Form jsou použity následující ovládací prvky (Controls):

- Menu, nabídka (MenuStrip) s třemi rozbalovacími nabídkami:
 - Soubor - Obsahuje následující položky nabídky:
 - Nový - Zresetuje Simulátor.
 - Otevřít - Spustí dialog pro výběr souboru.
 - Exit - Ukončení Simulátoru.
 - Úpravy - Obsahuje následující položky nabídky:
 - Stop - Pozastavení simulace.
 - Pokračovat - Spustit (pokračovat v) simulaci.
 - Další krok - Vykona se následující instrukce v simulaci.
 - Periferie - Rozbalí podnabídku s následujícími položkami :
 - Pole LED - Aktivuje okno s polem LED diod (Form LEDpole).
 - Displej - Aktivuje okno s numerickým displejem (Form Displej).
 - Klávesnice - Aktivuje okno s numerickou klávesnicí (Form Klavesnice).
 - Všechny periferie - Aktivuje Pole LED, Displej i Klávesnici.
 - Nastavení - Aktivuje okno s nastavením intervalu mezi instrukcemi v simulaci (Form FormNastaveni).
 - Nápověda - Obsahuje následující položky nabídky:
 - Nápověda - Aktivuje okno s popisem a nápovědou k ovládání Simulátoru (Form FormNapoveda).
 - AVR Assembler - Aktivuje okno se seznamem instrukcí AVR Assembleru (Form FormAVRassemblerNapoveda).
 - O aplikaci - Aktivuje okno s informacemi o Simulátoru (Form AboutBox1).

- Panel nástrojů (ToolStrip) s tlačítky (Button) pro (rychlý) výběr položek obsažených v menu.
Umožňuje výběr následujících (výše popsaných) položek :
 - Nový.
 - Otevřít.
 - Stop.
 - Pokračovat.
 - Další krok.
 - Pole LED.
 - Displej.
 - Klávesnice.
 - Nastavení.
 - AVR Assembler.
- Skupinové boxy (GroupBox) pro sdružení zobrazovaných informací (obsahy registrů a portů v AVR procesoru).
Použity pro sdružení :
 - groupBoxRegistry - zobrazení názvů a obsahů pracovních registrů.
 - groupBoxSRegs - zobrazení názvů a obsahů speciálních registrů (FLAGs I, T, H, S, V, N, Z, C.).
 - groupBoxPorty - zobrazení názvů a obsahů portů.
- Tabulky (TableLayoutPanel) pro rozvržení popisků v pravidelných rozestupech.
Umožňuje pravidelné rozložení zobrazených názvů a obsahů pracovních a speciálních registrů a portů.
- Popisky (Label) pro popis, jaké informace se zobrazují a některé hodnoty zobrazovaných informací (názvy registrů, portů, seznamů, hodnoty uložené v registrech a portech).
- Seznamy (ListBox) pro zobrazení obsahu paměti (RAM a Flash) a také pro zobrazení kódu v AVR Assembleru (tři samostatné seznamy).
- Časovač (Timer) pro vykonávání instrukcí kódu (programu) pro simulaci Simulátorem.
- Dialog (OpenFileDialog) pro výběr souboru s kódem v AVR Assembleru.

4.3.2. LEDpole

V této Windows Form jsou použity následující ovládací prvky (Controls):

- Popisky (Label) pro názvy portů a označení konkrétního bitu portu.
- Dioda - Vlastní ovládací prvek měnící barvu dle hodnoty bitu příslušného portu, odvozen z ovládacího prvku Obrázek (PictureBox).

4.3.3. Displej

V této Windows Form jsou použity následující ovládací prvky (Controls):

- Popisky (Label) pro názvy portů
- Obrázky (PictureBox) v podobě číslic pro zobrazení hodnot na portech (8 číslic, 2 pro každý port, zobrazovány hex. znaky "0" až "F").

4.3.4. Klavesnice

V této Windows Form jsou použity následující ovládací prvky (Controls):

- Popisek (Label) "Připojeno k portu".
- Přepínač (RadioButton) - mění port, ke kterému je klavesnice připojena.
- Tlačítka (Button) - 16 tlačítek s hex. znaky ("0" až "F").

4.3.5. FormNastaveni

V této Windows Form jsou použity následující ovládací prvky (Controls):

- Popisky (Label) - popisek "Nastavení prodlevy" a popisky s údaji o délce prodlevy (nad možnými polohami jezdce posuvníku).
- Posuvník (TrackBar) sloužící k nastavení délky prodlevy mezi vykonáním další instrukce kódu (programu) Simulátorem.
- Tlačítka (Button) - potvrzující tlačítka.

4.3.6. FormNapoveda

V této Windows Form je použit ovládací prvek (Control) Webový prohlížeč (WebBrowser). Zobrazí se v něm nápověda k ovládaní Simulátoru.

4.3.7. FormAVRassemblerNapoveda

V této Windows Form je použit ovládací prvek (Control) Webový prohlížeč (WebBrowser). Zobrazí se v něm seznam instrukcí (zkratky, možné hodnoty jejich parametrů, názvy instrukcí, popis operace, jaké příznaky (FLAGS) jsou operací ovlivněny).

4.3.8. AboutBox1

V této Windows Form jsou použity následující ovládací prvky (Controls):

- Obrázek (PictureBox) mikroprocesoru AVR.
- Popisky (Label) s informacemi o programu (Název, verze, licence, autor).
- Popisek s odkazem (LinkLabel) - e-mailová adresa.
- Tlačítko (Button) - potvrzující tlačítko.

4.4. Sestavení aplikace

K sestavení aplikace jsou dále potřeba html soubory pro zobrazení nápovědy ve FormNapoveda a FormAVRassemblerNapoveda. Při tvorbě html souboru s nápovědou k AVR Assembleru čerpal z popisu instrukční sady v [1] a [3]. Zdrojové kódy a soubory potřebné k sestavení aplikace v MS Visual C# jsou k dispozici na CD v příloze.

Závěr

Simulátor AVR je aplikace pro simulaci činnosti mikroprocesoru AVR. Umožňuje vykonávání vloženého programu po krocích a zobrazuje přitom údaje :

- Vykonávanou instrukci a instrukce v jejím okolí (v jazyce mikroprocesoru i v AVR Assembleru).
- Obsah obecných registrů.
- Obsah speciálních registrů.
- Obsah paměti RAM.
- Úroveň signálů na portech.

K portům je možné připojit zobrazovací jednotku v podobě pole LED diod, numerický displej a numerickou klávesnici.

Simulátor AVR má přehledné a jednoduché grafické uživatelské rozhraní, což uživateli umožňuje rychlou orientaci při práci s aplikací. Také vestavěná nápověda s popisem AVR Assembleru usnadní uživateli tvorbu programů v AVR Assembleru pro mikroprocesory AVR. Tyto programy je možno jednoduše vložit do Simulátoru AVR a sledovat běh programů či ověřit správnost kódu programu.

Simulátor AVR by mohl najít uplatnění zejména u uživatelů, kteří začínají s programováním mikroprocesorů AVR či hledají aplikaci s jednoduchým a přehledným uživatelským rozhraním pro kontrolu kódu při tvorbě programů pro mikroprocesory AVR.

V budoucích verzích Simulátoru uvažuji o některých dalších rozšířeních.

Mohly by to být například :

- Možnost přímé editace načteného kódu AVR Assembleru přímo v Simulátoru a uložení vytvořeného / upraveného kódu do souboru.
- Rozšíření předvoleb o volbu typu / řady mikroprocesoru AVR, který/á simulaci provádí, s čímž souvisí výběr ze sad instrukčních souborů.
- Rozšíření o další typy vstupních a výstupních periférií.
- Možnost sledování dalších údajů (například zásobník a další).

Reference

- [1] Váňa, Vladimír. *ATMEL AVR - Popis mikroprocesoru a instrukční soubor.*
BEN - technická literatura, Praha, 2003.
- [2] Váňa, Vladimír. *ATMEL AVR - ASSEMBLER.*
BEN - technická literatura, Praha, 2003.
- [3] Atmel Corporation. *ATmega8515(L) Summary.*
Elektronická dokumentace v pdf, 2006.
- [4] Atmel Corporation. *Atmel AVR 8-bit and 32-bit microcontroller.*
Webové stránky Atmel Corporation.

A. Obsah přiloženého CD

V samotném závěru práce uvádím popis obsahu přiloženého CD.

`bin/`

Instalátor SIMULÁTOR MIKROPROCESORU AVR programu.

`doc/`

Dokumentace práce ve formátu PDF, a dále všechny soubory nutné pro vygenerování PDF souboru dokumentace včetně zdrojového textu dokumentace (v ZIP archivu).

`src/`

Kompletní zdrojové texty programu SIMULÁTOR MIKROPROCESORU AVR.

`readme.txt`

Instrukce pro instalaci a spuštění programu SIMULÁTOR MIKROPROCESORU AVR, včetně požadavků pro jeho provoz.

Navíc CD obsahuje:

`data/`

Ukázkové vstupní soubory s kódy v AVR Assembleru pro program SIMULÁTOR MIKROPROCESORU AVR.