Dokumentace jednoduchého procesoru

Toto je dokumentace jednoduchého procesoru, jehož omezení jsou pečlivě vybrána, aby pomohla začátečníkům naučit se programovat v jazyce assembly. Doufám, že vám pomůže udělat první krok ke složitějším platformám :)

:: Technická specifikace

Displej: 160x144, fixní paleta s 16 barvami
Vstup: 7 tlačítek, sériový
Velikost pamětí (program/operační/rozhraní): 65536B/65536B/256B
Frekvence: 16Mhz

:: Syntaxe

Základním stavebním kamenem zdrojového kódu je řádka v následujícím tvaru:

```
    [návěstí:][mnemonika [operandy]][; Komentář]
```

Operandem může být:

- Návěstí (např. `Main')
- 8/16-bitová konstanta (např. 'a', 42, 0xDEAD, 0b1010, ...)
  - 8-bitovou konstantu lze získat i z adresy pomocí funkcí `hi' a `lo'. (např. `hi(Main)', `lo(loop)')
- Pro instrukci `db' pak ještě ASCII řetězec (např. "Hello!")
- Registr (např. r0, R15, ...)

Pro přehlednost bude dále používat následující zkratky:

- imm8/16: 8/16-bitová hodnota
- STR: Řetězec
- A: Adresa (návěstí nebo imm16)

:: Pseudoinstrukce

Překladač nabízí následující pseudoinstrukce pro ovlivnění překladu:

- `include <file>': Vloží obsah souboru <file> na místo užití
- `org imm16': Nastaví pozici v binárním souboru
- `db [imm8, STR]*': Vloží binární podobu operandů
- `ds imm16`: Posune kurzor o určitý počet bytů

:: Registry

Procesor disponuje následujícími registry:

- R0-R15: Víceúčelový registr
- X: Dvojregistr R12:R13, cíl instrukce `st`
- Y: Dvojregistr R14:R15, cíl instrukcí `ld` a `lpm`
- SP: Ukazatel na vrchol zásobníku
- PC: Čítač programu
- SR: Registr příznaků

Pouze registry R0-R15 lze používat jako operandy.

:: Příznaky

Registr příznaků obsahuje následující příznaky:

- I: Povolení přerušení
- Z: Nulový příznak
- C: Přenos do vyššího řádu

Všechny příznaky jsou nastavitelné pomocí patřičných instrukcí. Příznaky Z a C jsou nastaveny automaticky každou aritmetickou operací a lze je využívat jako podmínky pro skok.

:: Přerušení

Před načtením instrukce může dojít k následujícím přerušením, které skočí na patřičný vektor:

- VBlank: Obnovení obrazovky
- Button: Stisknutí tlačítka
- Serial: Příchozí zpráva na sériovém vstupu.

:: Rozdělení paměti

Procesor disponuje třemi druhy paměti. Každá paměť je adresována separátně. Adresní prostor každé paměti je popsán níže:

[Programová paměť]
  0x0000 – 0x0010: Reset vektor
  0x0010 – 0x0020: VBlank vektor
  0x0020 – 0x0030: Tlačítkový vektor
  0x0030 – 0x0040: Nevyužito
  0x0040 – 0x0050: Sériový vektor
  0x0050 – 0x0100: Nevyužito
  0x0100 – 0xFFFF: Uživatelský program

[Operační paměť]
  0x0000 – 0x8000: Uživatelská data
  0x8000 – 0xDA00: VRAM
  0xDA00 – 0xFFFF: Uživatelská data, zásobník

[Paměť rozhraní]
  0x02: Stisknutá tlačítka / –
  0x03: Fronta sériového vstupu / sériový výstup

:: Sériová linka

Procesor disponuje sériovou linkou, po které je schopen přijímat a odesílat zprávy v kódování ASCII. Pokud dojde k přerušení 'Serial', lze příchozí data číst byte po bytu z příslušné adresy rozhraní.

:: Displej

Emulátor disponuje malým displejem s rozměry 160x144 pixelů, jehož obsah odpovídá adresnímu prostoru operační paměti označenému jako 'VRAM'. Spodní čtyři bity každé hodnoty v rozsahu určují barvu pixelu jako index do fixní palety s 16 barvami. Jednotlivé pixely jsou namapovány po řádkách počínaje levým horním rohem.

:: Tlačítka

Emulátor disponuje sedmi tlačítky. V paměti rozhraní je na patřičné adrese uložen jejich stav. Při stisknutí tlačítka se provede přerušení 'Button'.

:: Instrukční sada

V této sekci si představíme celou instrukční sadu procesoru, včetně její binární podoby. Pro zkrácení budeme operandy instrukcí indikovat následovně:

- rX: Registr X
- i8/16: 8/16-bitová hodnota
- A: 16-bitová adresa

:: Strojový kód

Instrukce jsou reprezentovány vždy svým operačním znakem a binární podobou operandů, která je pro registry rozhodnuta následovně:

- Registr rX: 0x0X
- Registry rX, rY: 0xXY

Ostatní hodnoty a adresy jsou reprezentovány svou kanonickou binární podobou.

:: Instrukce

Instrukce jsou nadepsány ve formátu:

  [operační znak] mnemonika [operandy]: <popis>

[0x00] nop: Neprovede žádnou operaci.
[0x02] sleep: Přeruší chod procesoru do dalšího přerušení.
[0x03] break: Indikuje pozici breakpointu pro ladící program.

[0x04] sei: Nastaví příznak I.
[0x05] sec: Nastaví příznak C.
[0x06] sez: Nastaví příznak Z.
[0x07] cli: Odnastaví příznak I.
[0x08] clc: Odnastaví příznak C.
[0x09] clz: Odnastaví příznak Z.

[0x10] add rA, rB: Přičte k registru A hodnotu registru B.
[0x11] adc rA, rB: Přičte k registru A hodnotu registru B a přenos.
[0x12] sub rA, rB: Odečte od registru A hodnotu registru B.
[0x13] sbc rA, rB: Odečte od registru A hodnotu registru B a přenos.

[0x14] inc rA: Zvýší registr A o 1.
[0x15] dec rA: Sníží registr A o 1.

[0x16] and rA, rB: Přidá k registru A hodnotu registru B operací AND.
[0x17] or rA, rB: Přidá k registru A hodnotu registru B operací OR.
[0x18] xor rA, rB: Přidá k registru A hodnotu registru B operací XOR.

[0x19] cp rA, rB: Nastaví C pokud rX < rY, nastaví Z pokud rX == rY.
[0x1A] cpi rA, i8: Nastaví C pokud rX < i8, nastaví Z pokud rX == i8.

[0x20] jmp A: Nastaví PC na A.
[0x21] call A: Umístí PC na zásobník a nastaví PC na A.
[0x22] ret: Načte PC ze zásobníku.
[0x23] reti: Načte PC ze a nastaví příznak I.

[0x24] brc A: Nastaví PC na A pokud je nastaven příznak C.
[0x25] brnc A: Nastaví PC na A pokud je odnastaven příznak C.
[0x26] brz A: Nastaví PC na A pokud je nastaven příznak Z.
[0x27] brnz A: Nastaví PC na A pokud je odnastaven příznak Z.

[0x30] mov rA, rB: Zkopíruje do registru A hodnotu registru B
[0x31] ldi rA, i8: Uloží do registru A hodnotu i8

[0x32] ld rA: Načte do registru A z operační paměti hodnotu na adrese Y.
[0x36] lpm rA: Načte do registru A z programové paměti hodnotu na adrese Y.
[0x33] st rA: Uloží do operační paměti na adresu X hodnotu registru A.

[0x34] push rX: Umístí hodnotu rX na zásobník a inkrementuje SP.
[0x35] pop rX: Umístí do rX vrchol zásobníku a dekrementuje SP.

[0x3A] in rA, IO: Načte do registru A hodnotu z adresy IO paměti rozhraní.
[0x3B] out rA, IO: Uloží do paměti rozhraní na adresu IO hodnotu registru A.