

Návrh počítačových systémů 2006: Projekt č. 2

Rozšíření jednoduchého procesoru o šifrovací instrukci

Petr Zemek, xzemek02

1.

čas [ns]	ABUS	DBUS
620	110h	123
2060	111h	19B
3840	112h	215
5620	113h	27A
7400	114h	2E7
9180	115h	34C
10960	116h	3B7
12740	117h	3E7
14520	118h	419

login		x	z	e	m	e	k	0	2	
hexa (ASCII)		78	7A	65	6D	65	6B	30	32	
adresa		0110h	0111h	0112h	0113h	0114h	0115h	0116h	0117h	0118h
výsledek		023	09B	115	17A	1E7	24C	2B7	2E7	319

2.

```
when X"A" => ireg_dec <= madd;
--== zde doplnte vas kod ==
-- instrukcni dekodér
when X"E" => ireg_dec <= menc;
--== konec ==
when others => ireg_dec <= halt;
```

--ALU

with alu_oper select

```
alu_out <= alu_op0 + alu_op1 when alu_add,
alu_op1(7 downto 0)&alu_op1(15 downto 8) when alu_swap,
--== zde doplnte vas kod ==
enc_out(15 downto 0) when alu_enc,
--== konec ==
alu_op1 when others;
```

--ENCODING instruction

enc_out(7 downto 0) <=

--== zde doplnte vas kod ==

--pouzijte pouze signaly alu_op0 a alu_op1

((alu_op0(7 downto 0) xor alu_op1(7 downto 0)) and "01111110") xor alu_op0(15 downto 8)

--== konec ==

;

enc_out(15 downto 8) <=

--== zde doplnte vas kod ==

--pouzijte pouze signaly alu_op0, alu_op1 a enc_out(7 downto 0)

((alu_op1(15 downto 8) xor enc_out(7 downto 0)) and "01111110") xor alu_op0(7 downto 0)

--== konec ==

;

```
--== zde doplnte vas kod ==  
-- dekodovani instrukce  
when menc =>  
  nstate <= smenc0;  
--== konec ==  
--== zde doplnte vas kod ==
```

```
-- implementace instrukce  
-- ENCODE  
when smenc0 => --phase 0  
  nstate <= smenc1;  
  ireg_abus <= '1';  
  EN <= '1';  
  
when smenc1 => --phase 1  
  nstate <= sfetch0;  
  alu_oper <= alu_enc;  
  accmx_sel <= "10";  
  alu_mx1_sel <= '0';  
  alu_mx2_sel <= '0';  
  acc_ld <= '1';  
  pc_inc <= '1';  
--== konec ==
```

3.

login		x	z	e	m	e	k	0	2	
hexa (ASCII)		78	7A	65	6D	65	6B	30	32	
adresa		0110h	0111h	0112h	0113h	0114h	0115h	0116h	0117h	0118h
výsledek		0123	225A	7820	5944	1D28	354C	7926	5E16	4824

4.

