Simple-As-Possible Computers

Table of Contents

SAP-1

Control Signals

Old name	New name
C_P	inc
E_P	pc_out_en
$\sim L_M$	~ld_mar
~CE	~mem_out_en
$\sim L_I$	~ld_ir
$\sim E_I$	~ir_out_en
$\sim L_A$	~ld_acc
E_A	acc_out_en
S_U	sub_add
E_U	subadd_out_en
$\sim L_B$	~ld_b_reg
$\sim L_O$	~ld_out_reg

T-state active signals

- **T_1 state** (Address state)
  - `pc_out_en`
  - `~ld_mar`
- **T_2 state** (Increment state)
  - `inc`
- **T_3 state** (Memory state)
  - `~mem_out_en`
  - `~ld_ir`
- `LDA` routine
  - **T_4 state**
    - `~ir_out_en`
    - `~ld_mar`
  - **T_5 state**
    - `~mem_out_en`
    - `~ld_acc`

    - **T_6 state**
        - NOP
- `ADD` routine
    - **T_4 state**
        - `~ir_out_en`
        - `~ld_mar`
    - **T_5 state**
        - `~mem_out_en`
        - `~ld_b_reg`
    - **T_6 state**
        - `subadd_out_en`
        - `~ld_acc`
- `SUB` routine
    - **T_4 state**
        - `~ir_out_en`
        - `~ld_mar`
    - **T_5 state**
        - `~mem_out_en`
        - `~ld_b_reg`
    - **T_6 state**
        - `sub_add`
        - `subadd_out_en`
        - `~ld_acc`
- `OUT` routine
    - **T_4 state**
        - `acc_out_en`
        - `~ld_out_reg`
    - **T_5 state**
        - `~NOP`
    - **T_6 state**
        - `NOP`

Control logic

```
inc = t2;
pc_out_en = t1;
~ld_mar = ~(t1 || (t4 && (lda || add || sub)));
~mem_out_en = ~(t3 || (t4 && (lda || add || sub)));
~ld_ir = ~t3;
~ir_out_en = ~(t4 && (lda || add || sub));
~ld_acc = ~((t5 && lda) || (t6 && (add || sub)));
acc_out_en = t4 && out;
sub_add = t6 && sub;
subadd_out_en = t6 && (add || sub);
~ld_b_reg = ~(t5 && (add || sub));
~ld_out_reg = ~(t4 && out);
```