**SIM1 TO SIM5 PROGRAMMING CARD**

**acc the 4 digit accumulator**

**ip the 3 digit instruction pointer**

**aaa a 3-digit memory address**

**M[aaa] the 4 digits in memory address aaa**

**dddd 4-digit number for input or output**

**symb a symbolic address (e.g. loop, cnt, j)**

 **CPU AND MEMORY THE INSTRUCTION CYCLE**

**+----------CPU---------+ +-----------------+**

**| +----+ +---+ | +-->|inst = memory[ip]|**

**| acc |0000| ip |000| | | +-----------------+**

**| +----+ +---+ | | |**

**+----------------------+ | v**

 **/ \ | +-----------------+**

 **/| |\ | | ip = ip + 1 |**

 **| | | +-----------------+**

 **\| |/ | |**

 **\ / | v**

 **+------------+ | +-----------------+**

 **| addr cont | | | execute inst |**

 **| +----+ | | +-----------------+**

 **| 000 |1004| | | |**

 **| +----+ | | v**

 **| 001 |3005| | <-------------**

 **| +----+ |**

 **| 002 |2006| |**

 **| +----+ |**

 **| ... ... |**

 **| +----+ |**

 **| 998 |0000| |**

 **| +----+ |**

 **| 999 |0000| |**

 **| +----+ |**

 **+---MEMORY---+**

**MEMORY INSTRUCTIONS (SIM1)**

 **machine assembly**

**name format format effect**

**HALT 0000 halt halt the processor**

**LD 1aaa ld symb acc = M[aaa]**

**ST 2aaa st symb M[aaa] = acc**

**ADD 3aaa add symb acc = acc + M(aaa)**

**SUB 4aaa sub symb acc = acc - M(aaa)**

**LDA 5aaa lda symb acc = 0aaa**

**JUMP AND SKIP INSTRUCTIONS (SIM2)**

 **machine assembly**

**name format format effect**

**JMP 6aaa jmp symb ip = aaa**

**SKIP 7000 skip ip = ip + 1**

**SKEQ 7100 skeq if (acc==0) ip=ip+1**

**SKNE 7200 skne if (acc!=0) ip=ip+1**

**SKGT 7300 skgt if (acc> 0) ip=ip+1**

**SKGE 7400 skge if (acc>=0) ip=ip+1**

**SKLT 7500 sklt if (acc< 0) ip=ip+1**

**SKLE 7600 skle if (acc<=0) ip=ip+1**

**ADDITIONAL ACCUMULATOR INSTRUCTIONS (SIM3)**

 **machine assembly**

**name format format effect**

**IN 8000 in acc = input device**

**OUT 8100 out output device = acc**

**CLR 8200 clr acc = 0000**

**INC 8300 inc acc = acc + 1**

**DEC 8400 dec acc = acc - 1**

**NEG 8500 neg acc = - acc**

**SHFTL 8600 shl acc = acc \* 10**

**SHFTR 8700 shr acc = acc / 10**

**SIGNED AND UNSIGNED NUMBERS**

**(use only skip, skeq and skne with unsigned)**

**memory unsigned signed**

 **0000 0 0**

 **0001 1 1**

 **0002 2 2**

 **0003 3 3**

 **... ... ...**

 **4998 4998 4998**

 **4999 4999 4999**

 **5000 5000 -5000**

 **5001 5001 -4999**

 **5002 5002 -4998**

 **... ... ...**

 **9996 9996 -4**

 **9997 9997 -3**

 **9998 9998 -2**

 **9999 9999 -1**

**SIM4 CPU REGISTERS (10 general registers)**

 **+------CPU------+**

 **| reg content |**

 **| +------+ |**

 **| 0 | dddd | | %r0 is %acc**

 **| +------+ |**

 **| 1 | dddd | |**

 **| +------+ |**

 **| ... |**

 **| +------+ |**

 **| 7 | 0ddd | | %r7 is %sp**

 **| +------+ |**

 **| 8 | 0ddd | | %r8 is %lr**

 **| +------+ |**

 **| 9 | 0ddd | | %r9 is %ip**

 **| +------+ |**

 **+------CPU------+**

**SIM4 PROGRAMMING CARD**

**s general register used as source (r0-r9)**

**d general register as destination (r0-r9)**

**rn general register n (0 to 9)**

**aaa a 3-digit memory address**

**M[aaa] the 4 digits in memory address aaa**

**R[x] the 4 digits in general register x**

**dddd 4-digit number**

**symb a symbolic address (e.g. loop, cnt, j)**

**MEMORY INSTRUCTIONS (uses %r0 only)**

 **machine assembly**

**name format format effect**

**HALT 0000 halt halt the processor**

**LD 1aaa ld symb R[0] = M[aaa]**

**ST 2aaa st symb M[aaa] = R[0]**

**ADD 3aaa add symb R[0] = R[0] + M(aaa)**

**SUB 4aaa sub symb R[0] = R[0] - M(aaa)**

**LDA 5aaa lda symb R[0] = 0aaa**

**JUMP AND SKIP INSTRUCTIONS (any general register)**

 **machine assembly**

**name format format effect**

**JMP 6aaa jmp symb R[9] = aaa**

**SKIP 7000 skip R[9] = R[9]+1**

**SKEQ 710s skeq %s if (R[s]==0) R[9]=R[9]+1**

**SKNE 720s skne %s if (R[s]!=0) R[9]=R[9]+1**

**SKGT 730s skgt %s if (R[s]> 0) R[9]=R[9]+1**

**SKGE 740s skge %s if (R[s]>=0) R[9]=R[9]+1**

**SKLT 750s sklt %s if (R[s]< 0) R[9]=R[9]+1**

**SKLE 760s skle %s if (R[s]<=0) R[9]=R[9]+1**

**SINGLE RESISTER INSTRUCTIONS (any general register)**

 **machine assembly**

**name format format effect**

**IN 800d in %d R[d] = input device**

**OUT 810s out %s output device = R[s]**

**CLR 820d clr %d R[d] = 0000**

**INC 830d inc %d R[d] = R[d] + 1**

**DEC 840d dec %d R[d] = R[d] - 1**

**NEG 850d neg %d R[d] = 0.0 - R[d]**

**SHFTL 860d shl %d R[d] = R[d] \* 10**

**SHFTR 870d shr %d R[d] = R[d] / 10**

**TWO REGISTER INSTRUCTIONS (any general register)**

 **machine assembly**

**name format format effect**

 **------> <-------**

**MVRR 90sd mvrr %s,%r R[d]=R[s]**

**MVMR 91sd mvmr (%s),%d R[d]=M[R[s]]**

**MVRM 92sd mvrm %s,(%d) M[R[d]]=R[s]**

**EXCH 93sd exch %s,%d R[d]<=>R[s]**

**ADDR 94sd addr %s,%d R[d]=R[d]+R[s]**

**SUBR 95sd subr %s,%d R[d]=R[d]-R[s]**

**SIM5 INSTRUCTIONS**

**PUSH 880s push %s M[--R[7]]=R[s]**

**POP 890d pop %d R[d]=M[R[7]++]**

**CALL 960s call %s push %r9,R[9]=R[s]**

**RET 8909 ret pop %r9**

**Load %r5 with content of memory cell "cnt"**

 **version 1 version 2**

 **ld cnt lda cnt**

 **mv %r0,%r5 mv (%r0),%r5**