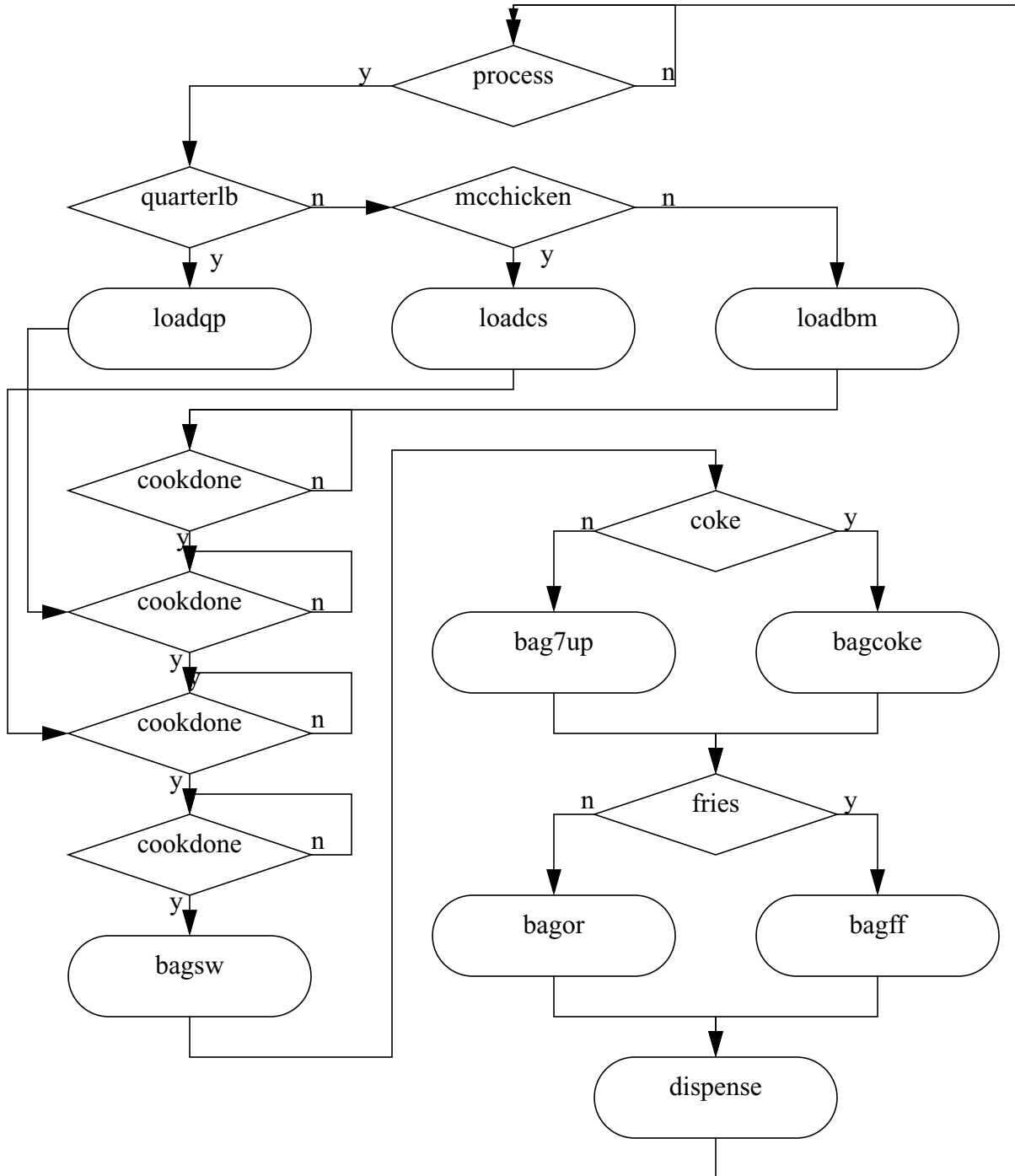


Massachusetts Institute of Technology  
 Department of Electrical Engineering and Computer Science  
 6.111 Introductory Digital Systems Laboratory  
 Problem Set #5 Solutions



```

/* McCU.as */
#SPEC_FILE = McCU.sp;
#LIST_FILE = McCU.lst;
#SET_ADDRESS = 0;

start:  if process jump sw;
        goto start;

sw:     if quarterlb jump qp;
        if mcchicken jump mckn;
        if bigmac jump bm;

qp:     cooker loadqp;
        goto wait3;

mckn:  cooker loadcs;
        goto wait2;

bm:     cooker loadbm;

wait4:  if cookdone jump wait3;
        goto wait4;

wait3:  if cookdone jump wait2;
        goto wait3;

wait2:  if cookdone jump wait1;
        goto wait2;

wait1:  if cookdone jump bag;
        goto wait1;

bag:    loader bagsw;

        if coke jump bgcoke;
        loader bag7up;
        goto ff;
bgcoke: loader bagff;

ff:     if fries jump bgff;
        loader bagor;
        goto dis;
bgff:   loader bagff;

dis:    loader dispense;
        goto start;

```

```

/* McCU.sp, spec file for McCU.as */

/* MCU instruction is 9-bits wide. */
op<8:0>;

/* The <address> field of instruction is bits 0 through 4. */
address op<4:0>;

/* Opcode b00: if <condition> jump <address> */
if op<8>=0;
jump nop;

goto op<8:5>=%b0111;

/* Opcode b10: cooker <cooker-signals> */
cooker op<8:7>=%b10;

/* Opcode b11: loader <loader-signals> */
loader op<8:7>=%b11;

/* Condition codes */
cookdone    op<7:5>=0;
bigmac      op<7:5>=1;
quarterlb   op<7:5>=2;
mchicken    op<7:5>=3;
fries       op<7:5>=4;
coke        op<7:5>=5;
process     op<7:5>=6;
true        op<7:5>=7;

/* Control lines */

loadbm      op<0>=1;
loadqp      op<1>=1;
loadcs      op<2>=1;
cooklmin    op<3>=1;

bagsw       op<0>=1;
bagor       op<1>=1;
bagff       op<2>=1;
bagcoke     op<3>=1;
bag7up      op<4>=1;
dispense    op<5>=1;

```

