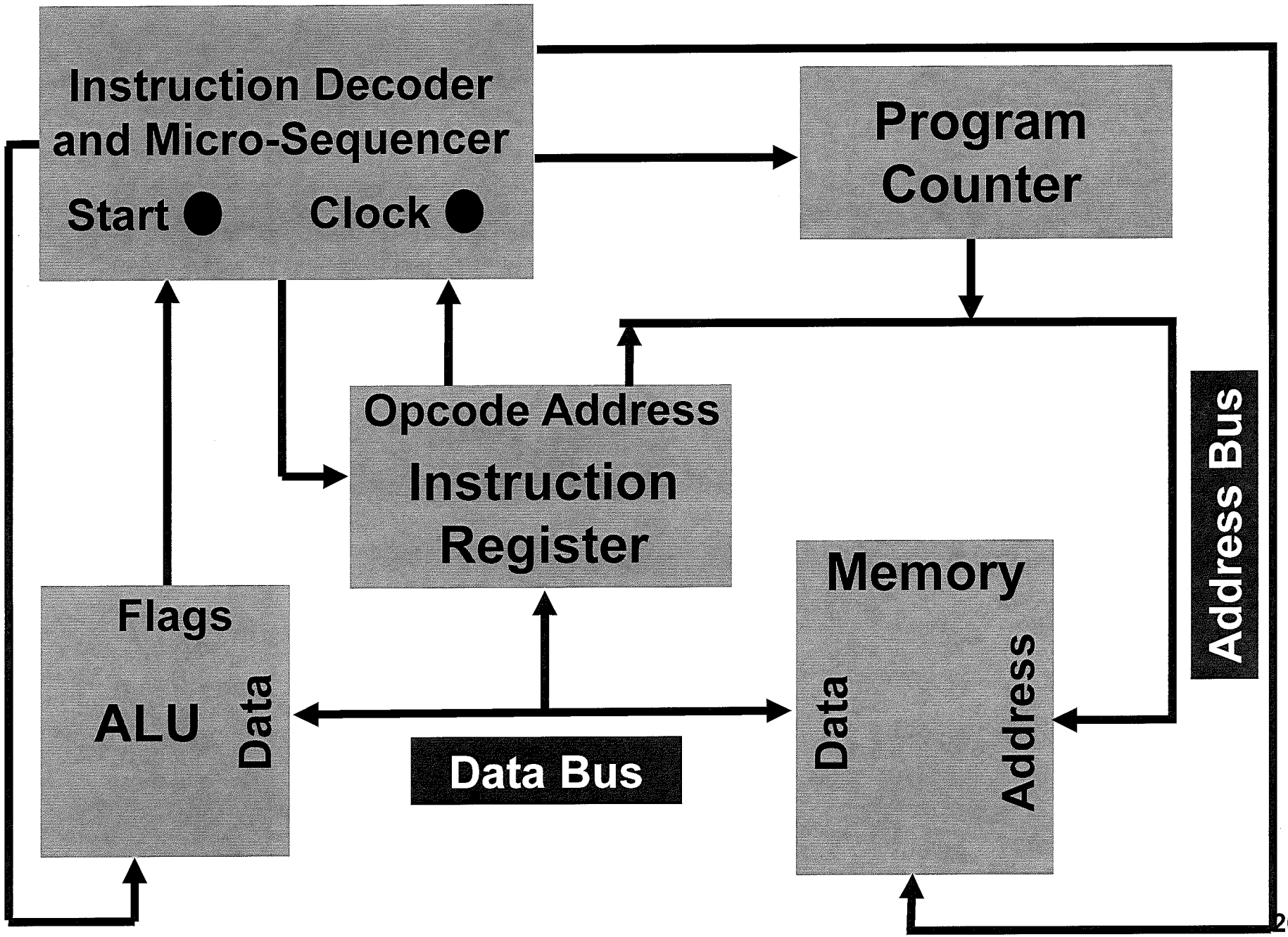
Instruction Decoder
and Micro-Sequencer
Start
Clock
Program
Counter
Opcode Address
Instruction
Register
Flags
ALU
Data
Data Bus
Memory
Data
Address
Address Bus

System Control Signals

Decoded State	Instruction Mnemonic	MSL	MOE	MWE	PCC	POA	IRL	IRA	AOE	ALE	ALX	ALY
S0	—	H	H		H	H	H					
S1	LDA	H	H					H		H	H	
S1	STA	H		H				H	H			
S1	ADD	H	H					H		H		
S1	SUB	H	H					H		H		H
S1	AND	H	H					H		H	H	H
S1	HLT	L			L		L			L		

```
MODULE idms

TITLE    'Instruction Decoder and Microsequencer'

DECLARATIONS

CLOCK pin;

START pin; " asynchronous START pushbutton

OP0..OP2 pin; " opcode bits (input from IR5..IR7)

" State counter
SQ node istype 'reg_D,buffer';

" RUN/HLT state
RUN node istype 'reg_D,buffer';

" Memory control signals
MSL,MOE,MWE pin istype 'com';

" PC control signals
PCC,POA,ARS pin istype 'com';

" IR control signals
IRL,IRA pin istype 'com';

" ALU control signals (not using flags yet)
ALE,ALX,ALY,AOE pin istype 'com';
```

```
" Decoded opcode definitions

LDA = !OP2&!OP1&!OP0;  " LDA opcode = 000
STA = !OP2&!OP1& OP0;  " STA opcode = 001
ADD = !OP2& OP1&!OP0;  " ADD opcode = 010
SUB = !OP2& OP1& OP0;  " SUB opcode = 011
AND =  OP2&!OP1&!OP0;  " AND opcode = 100
HLT =  OP2&!OP1& OP0;  " HLT opcode = 101

" Decoded state definitions

S0 = !SQ.q; " fetch
S1 =  SQ.q; " execute

EQUATIONS

" State counter
SQ.d = RUN.q&!SQ.q; " if RUN negated, resets SQ
SQ.clk = CLOCK;
SQ.ar = START;      " start in fetch state

" Run/stop (equivalent of SR latch)
RUN.ap = START;  " start with RUN set to 1
RUN.clk = CLOCK;
RUN.d = RUN.q;
RUN.ar = S1&HLT;  " RUN is cleared when HLT executed
```

```
" System control equations

MSL = RUN.q&(S0 # S1&(LDA # STA # ADD # SUB # AND));
MOE = S0 # S1&(LDA # ADD # SUB # AND);
MWE = S1&STA;
ARS = START;
PCC = RUN.q&S0;
POA = S0;
IRL = RUN.q&S0;
IRA = S1&(LDA # STA # ADD # SUB # AND);
AOE = S1&STA;
ALE = RUN.q&S1&(LDA # ADD # SUB # AND);
ALX = S1&(LDA # AND);
ALY = S1&(SUB # AND);

END
```