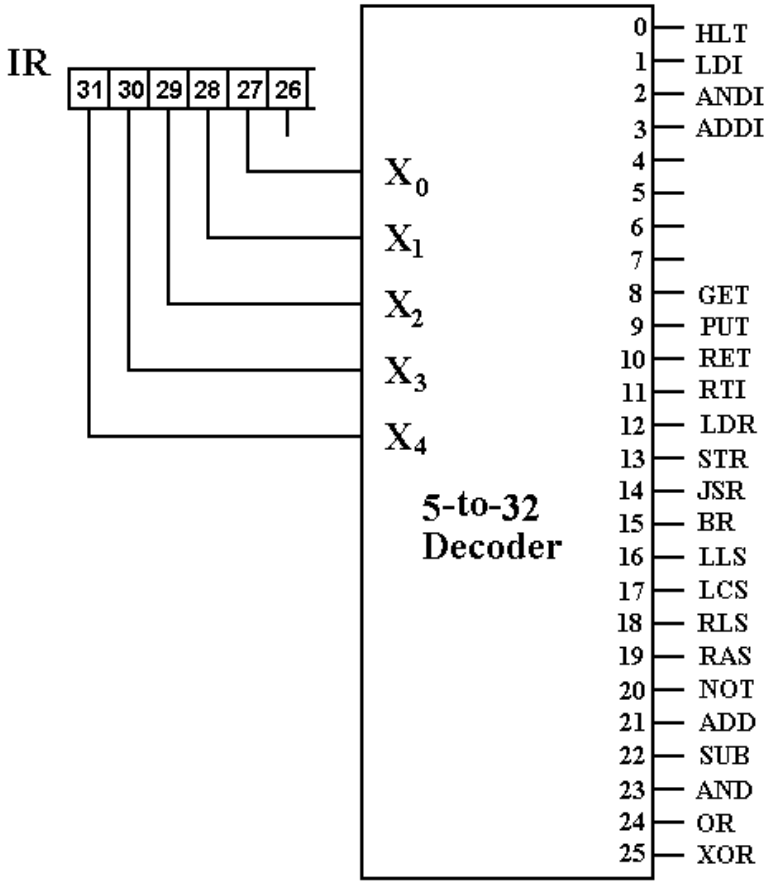
The Instruction Decoder

The instruction decoder, used by both the hardwired control unit and microprogrammed control unit, is a simple 5–to–32 active high decoder with some outputs not used.



The Signal Generation Trees

The hardwired control unit is implemented as a number of signal generation trees.

The input to each tree is as follows:

1. The discrete signal from the decoder indicating the instruction.
2. The major state: F, D, or E.
3. The minor state register: T0, T1, T2, or T3.

We should be careful about discussing the discrete outputs of the Instruction Decoder.

Each of these is a Boolean discrete signal with only two values: 0 or 1.

Example:

The JSR instruction has op-code = 14.

When this op–code is in the IR, decoder output Y14 = 1. We say JSR = 1.

There is a JSR instruction in the assembly language and a JSR discrete signal
in the microarchitecture that corresponds to the instruction, but is not identical to it.

The Common Fetch Sequence

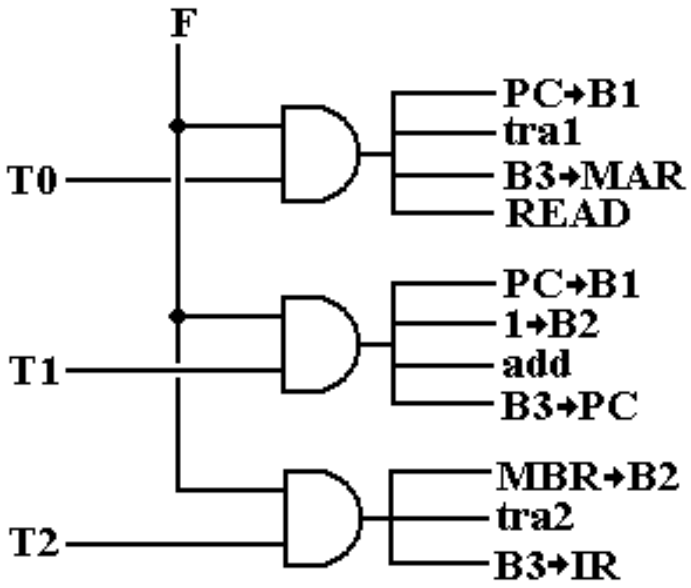
Here is the control signal sequence for the common fetch.

F, T0: PC → B1, **tra1**, B3 → MAR, READ. // MAR ← (PC)

F, T1: PC → B1, 1 → B2, **add**, B3 → PC. // PC ← (PC) + 1

F, T2: MBR → B2, **tra2**, B3 → IR. // IR ← (MBR)

Here is the signal generation tree for the common fetch.

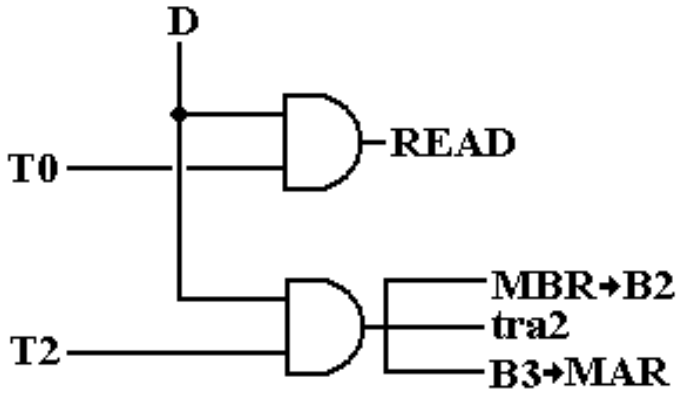


The Sequence for the Defer State

Here is the control signal sequence for the Defer State.

D, T0:	READ.	// Address is already in the MAR.
D, T1:	WAIT.	// Cannot access the MBR just now.
D, T2:	MBR $\rightarrow$ B2, **tra2**, B3 $\rightarrow$ MAR.	// MAR $\leftarrow$ (MBR)
D, T3:	WAIT.	// Effective Address is now in the MAR.

Here is the signal generation tree.



Fetch, T3

We summarize the actions in (F, T3) in order to generate signals.

Op–Code						B1	B2	B3	ALU	Other
IR_{31}	IR_{30}	IR_{29}	IR_{28}	IR_{27}						
0	0	0	0	0	HLT					0 → RUN
0	0	0	0	1	LDI	IR		R	tra1	
0	0	0	1	0	ANDI	IR	R	R	and	
0	0	0	1	1	ADDI	IR	R	R	add	
0	1	0	0	0	GET					
0	1	0	0	1	PUT					
0	1	0	1	0	RET					
0	1	0	1	1	RTI					
0	1	1	0	0	LDR	IR	R	MAR	add	
0	1	1	0	1	STR	IR	R	MAR	add	
0	1	1	1	0	JSR	IR	R	MAR	add	
0	1	1	1	1	BR	IR	R	MAR	add	
1	0	0	0	0	LLS		R	R	shift	1, 0, 0*
1	0	0	0	1	LCS		R	R	shift	1, 0, 1
1	0	0	1	0	RLS		R	R	shift	0, 0, 0
1	0	0	1	1	RAS		R	R	shift	0, 1, 0
1	0	1	0	0	NOT		R	R	not	
1	0	1	0	1	ADD	R	R	R	add	
1	0	1	1	0	SUB	R	R	R	sub	
1	0	1	1	1	AND	R	R	R	and	
1	1	0	0	0	OR	R	R	R	or	
1	1	0	0	1	XOR	R	R	R	xor	

Commonalities in Fetch, T3 (Part 1)

We begin to note a few commonalities in the (F, T3) control signals
that will help to simplify the signal generation tree.

If $IR_{31} = 1$, we have two distinct classes of instructions.

ADD, SUB, AND, OR, XOR

These dyadic instructions issue R $\rightarrow$ B1, R $\rightarrow$ B2, and B3 $\rightarrow$ R.
These are issued in addition to the control signal for the specific ALU operation.

LLS, LCS, RLS, RAS, NOT

These monadic instructions issue R $\rightarrow$ B2, and B3 $\rightarrow$ R.
These are issued in addition to the control signal for the specific ALU operation.

Note that we could issue R $\rightarrow$ B1 for these five monadic instructions as well.
This would likely put register %R0 on bus B1, but that bus would not be used.

SIMPLIFICATION

If $IR_{31} = 1$, the signal generation tree will generate these signals in addition to
the control signal for the specific ALU operation.

R $\rightarrow$ B1, R $\rightarrow$ B2, and B3 $\rightarrow$ R

Commonalities in Fetch, T3 (Part 2)

Here we note the commonalities for instructions with $IR_{31} = 0$.

All instructions that use bus B1 issue the control signal IR $\rightarrow$ B1.

We simplify the signal generation tree by causing
all instructions with $IR_{31} = 0$ to issue the signal IR $\rightarrow$ B1.

All instructions that use bus B2 issue the control signal R $\rightarrow$ B2.
This holds true for both $IR_{31} = 0$ and $IR_{31} = 1$.

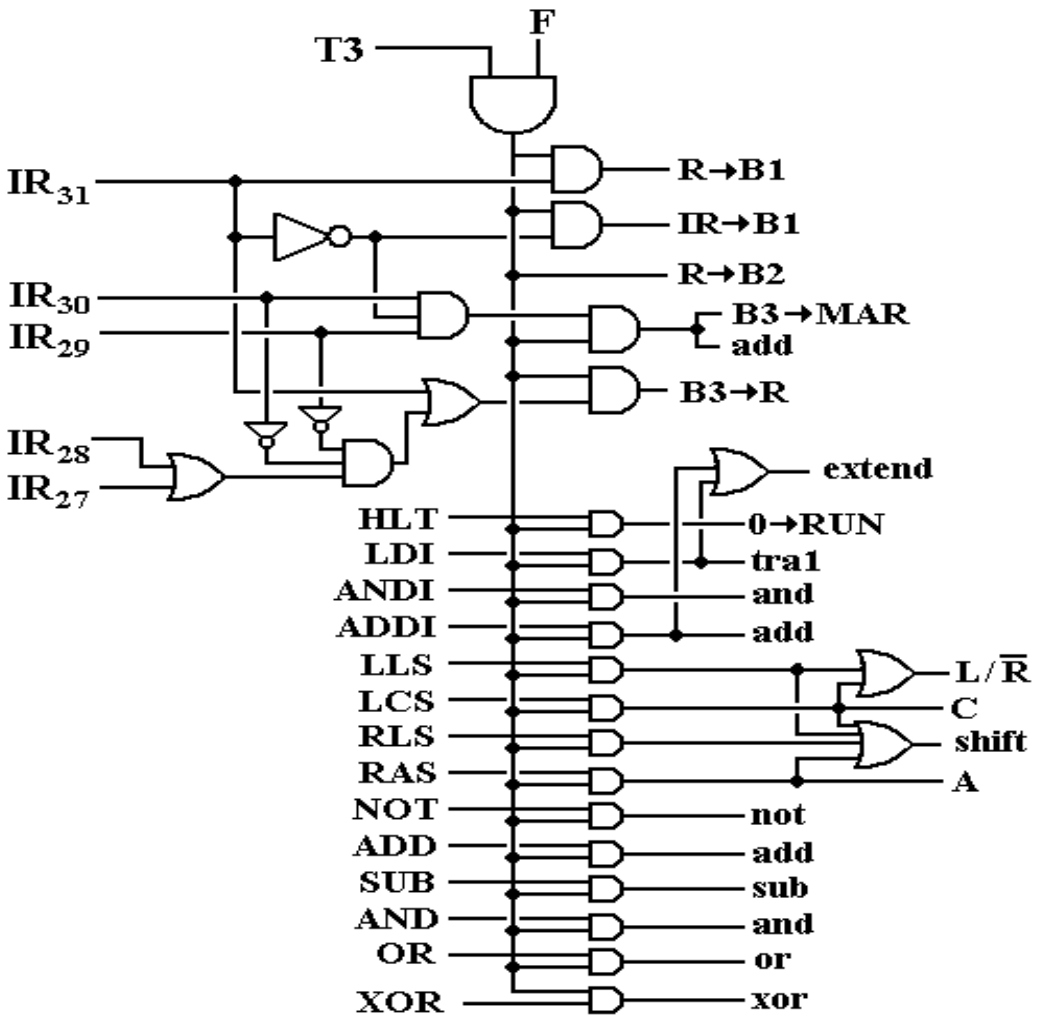
We just assert R $\rightarrow$ B2 whenever the control unit is in (Fetch, T3).

All instructions with $IR_{31} = 0$, $IR_{30} = 1$, and $IR_{29} = 1$ issue the control signals
B3 $\rightarrow$ MAR, add.

All instructions with $IR_{31} = 0$, $IR_{30} = 0$, and $IR_{29} = 0$ that use bus B3 issue the
control signal B3 $\rightarrow$ R.

Signal Generation Tree for (Fetch, T3)

Here it is. The handling of IR_{31} in the book is not correct.



Study for the Execute State (T0 and T1)

Here we compile all actions taken during Execute and organize by the Minor State and instruction.

Execute, T0

GET: IR → B1, **tra1**, B3 → IOA.

PUT: R → B2, **tra2**, B3 → IOD

RET: SP → B1, – 1 → B2, add, B3 → SP.

LDR: READ.

JSR: PC → B1, **tra1**, B3 → MBR.

Execute, T1

RET: SP → B1, tra1, B3 → MAR, READ.

STR: R → B1, **tra1**, B3 → MBR, WRITE.

JSR: MAR → B1, **tra1**, B3 → PC.

Study for the Execute State (T2 and T3)

Execute, T2

GET: IOD → B2, **tra2**, B3 → R.

PUT: IR → B1, **tra1**, B3 → IOA.

LDR: MBR → B2, **tra2**, B3 → R.

JSR: SP → B1, **tra1**, B3 → MAR, WRITE.

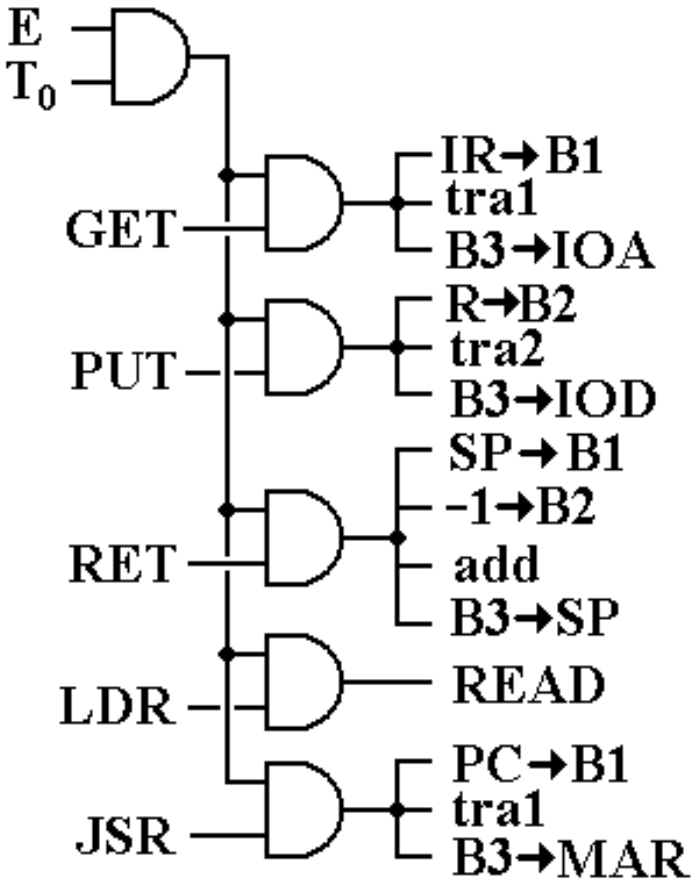
Execute, T3

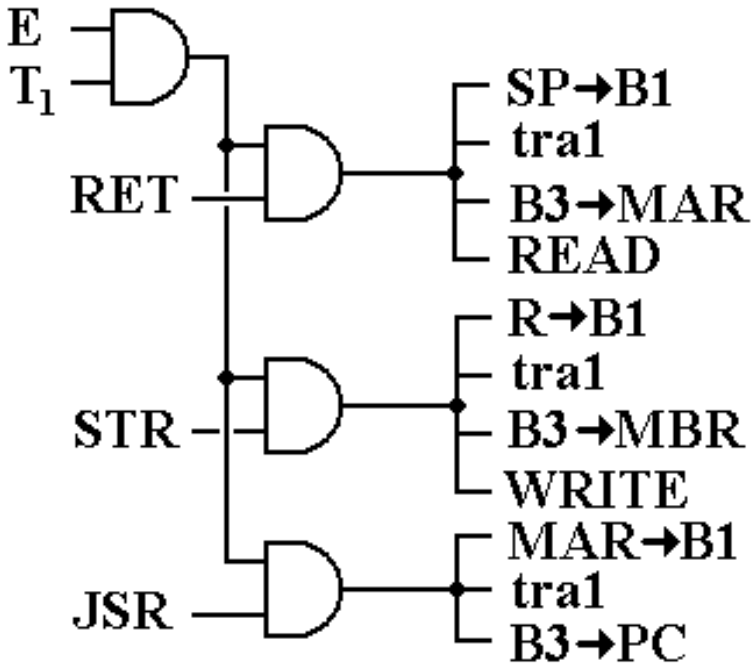
RET: MBR → B2, **tra2**, B3 → PC.

JSR: SP → B1, 1 → B2, **add**, B3 → SP.

BR: MAR → B1, **tra1**, B3 → PC.

Control Signals for the Execute State (T0 and T1)





Control Signals for the Execute State (T2 and T3)