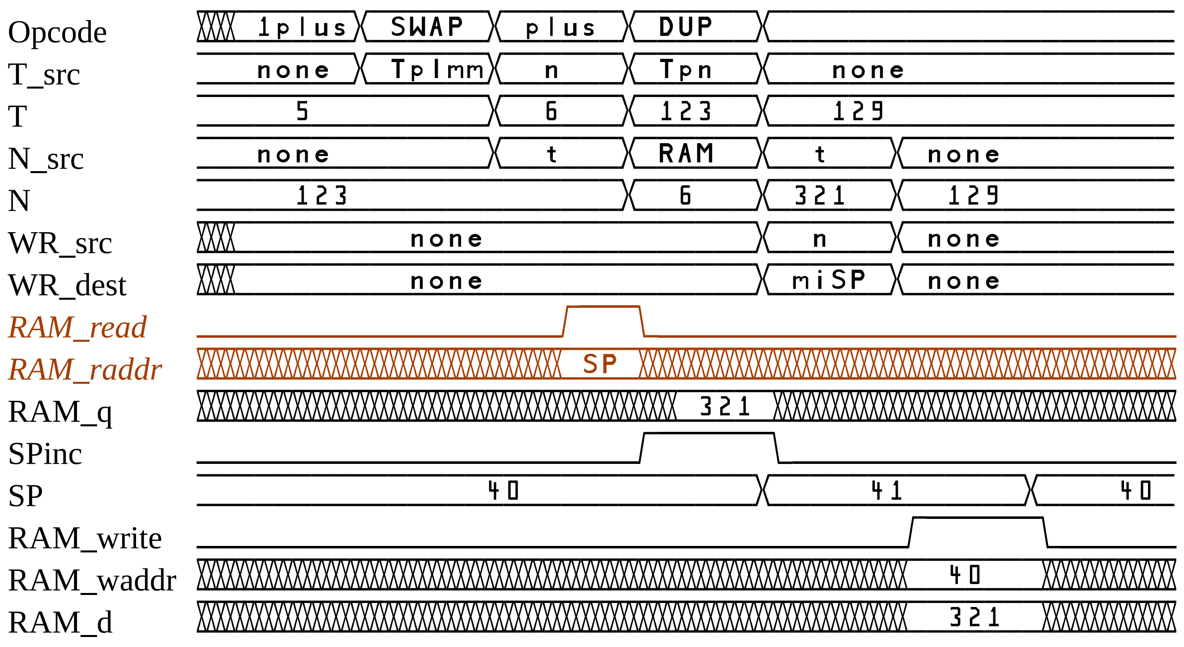
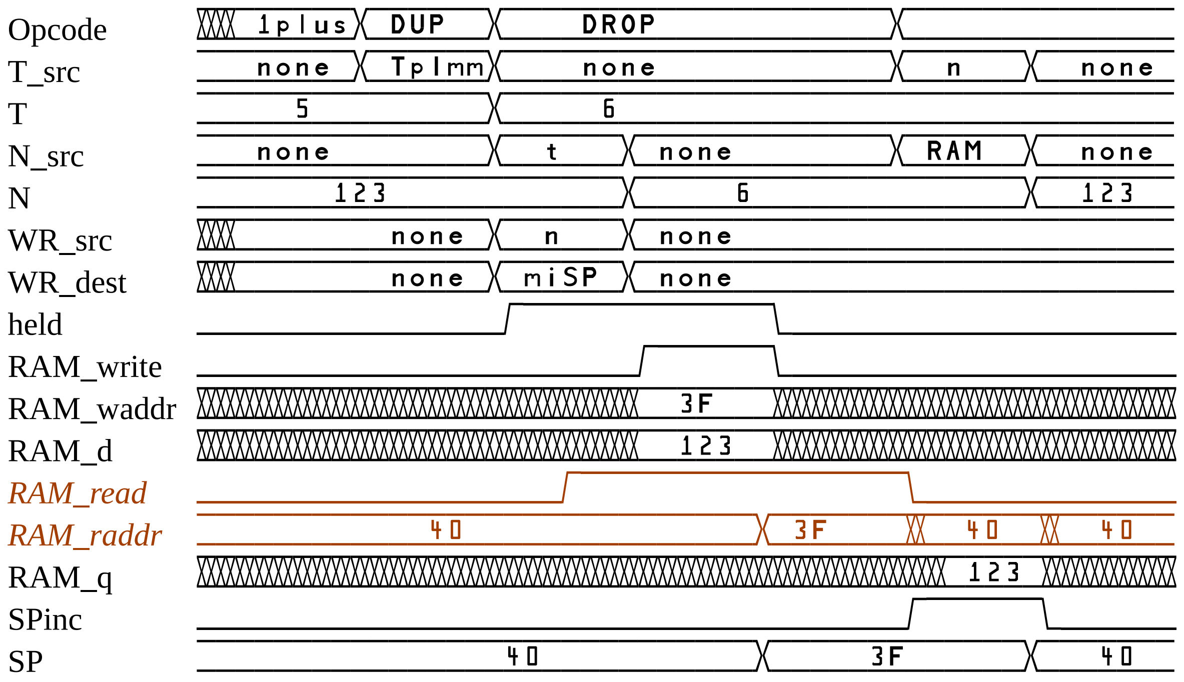
Block RAM

Block RAM uses synchronous read, whose signals are asynchronously decoded from **opcode**. Synchronous opcode decoding sets up the sources for various registers. For example, **new_T** indicates that **T** is to be loaded from 1 of 16 sources.

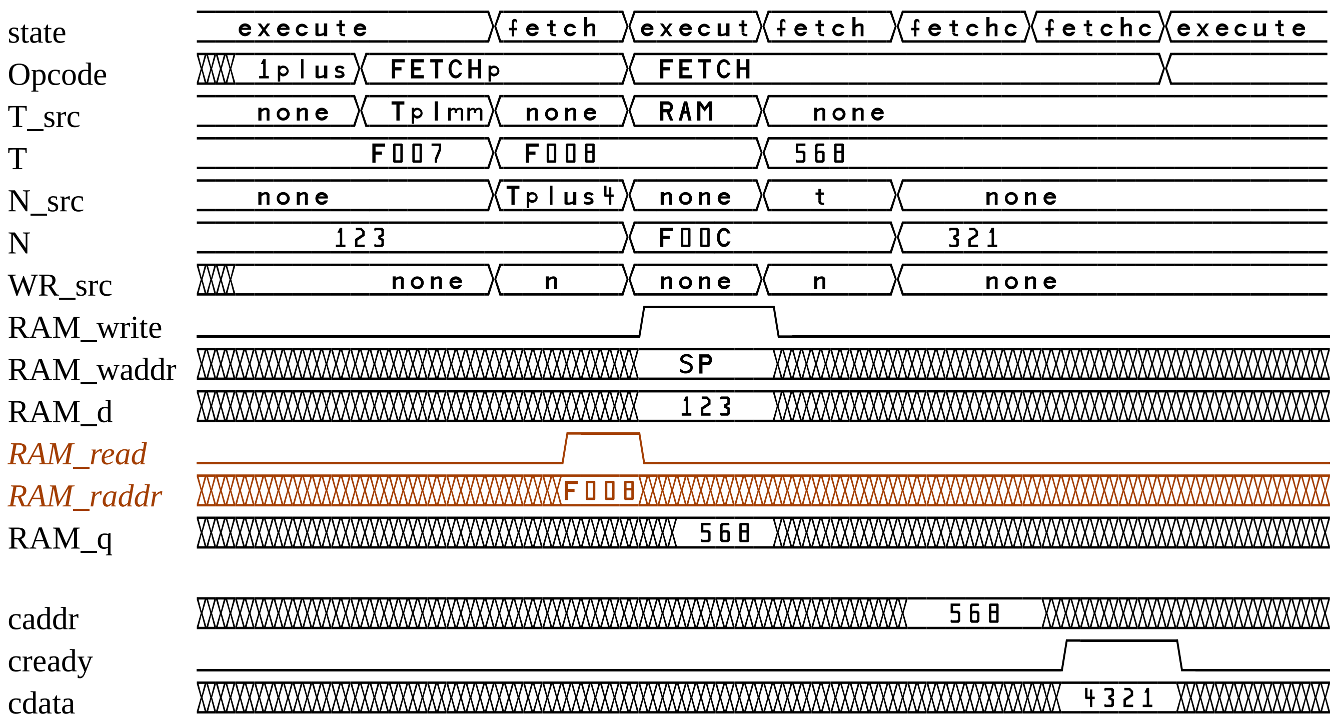
DROP followed by DUP:



DUP followed by DROP: To prevent DROP's read from occurring before DUP's write, "WR_src" must be "none" to allow decoding when is a read is expected. If single-port RAM is used, decoding must also be held off while **RAM_write** is high.

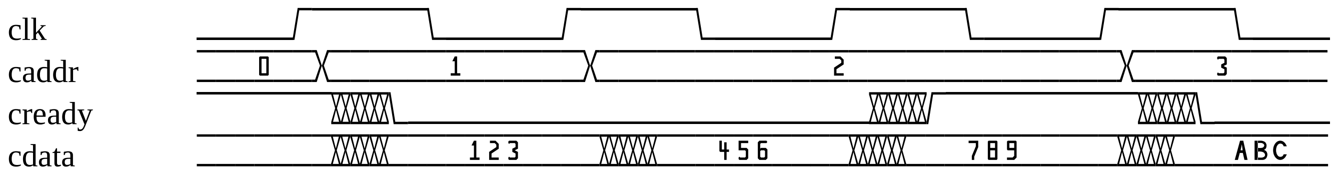


Fetch needs T before it can start a read. @+ (a – a+4 n).
The fetch state checks T and fetches from ROM space (instead) if necessary.


state: execute, fetch, execut, fetch, fetchc, fetchc, execute
Opcode: 1plus, FETCHp, FETCH
T_src: none, TpImm, none, RAM, none
T: F007, F008, 568
N_src: none, Tplus4, none, t, none
N: 123, F00C, 321
WR_src: none, n, none, n, none
RAM_write
RAM_waddr: SP
RAM_d: 123
RAM_read
RAM_raddr: F008
RAM_q: 568
caddr: 568
cready
cdata: 4321

ROM

The ROM interface uses CADDR, CDATA, and CREADY. It's the CPU's interface to large ROM. When CADDR changes, CREADY drops until CDATA matches the new data. Usually it's a synchronous-read ROM. A dumb version would not look ahead:



Fishbone Interface

I wanted an AXI4 interface, but 300 pages of documentation just to move data from point A to point B? One page is easier. A tiny mutant of AXI is used instead, enough to enable block transfers using far fewer signals. It can bridged to either Wishbone or AXI without too much trouble. If you're lazy or afraid of commitment, this may be the interface for you.

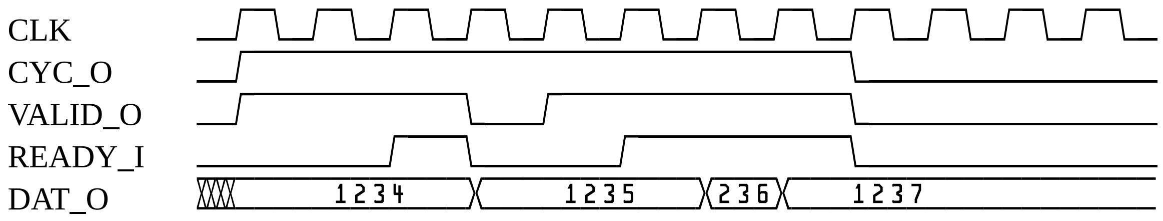
The T register is the read/write address and IMM is the burst length. These are output to the Fishbone bus directly. The bus signals are:

CYC_O	Trigger a read or write burst of IMM-1 words starting at byte address T. Drop after transfer.
WE_O	'1'=write, '0'=read. Steady throughout cycle.
BLEN_O	Burst length less 1, copy of IMM[7:0].
BADR_O	Block address, copy of T. T remains constant throughout the transfer.
VALID_O	AXI-type handshake for output.
READY_I	
DAT_O	Outgoing data, 32-bit.
VALID_I	AXI-type handshake for input.
READY_O	
DAT_I	Incoming data, 32-bit.

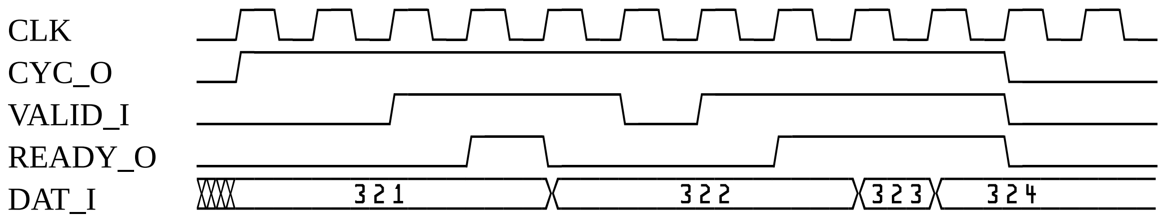
So, eight non-trivial signals instead of 40. A time-out of 4096 beats triggers an error interrupt if:
Output is hung: READY_I is stuck low instead of responding to VALID_O.
Input is hung: VALID_I is stuck low instead of responding to READY_O.

A Fishbone slave should implement a FSM that waits for CYC_O to rise to start a burst and waits for it to fall when finished. The master is expected to provide enough gap for the FSM to return to idle. WE_O, BLEN_O, and BADR_O are expected to be stable throughout CYC_O. The slave FSM only needs them at the leading edge of CYC_O. There are two types of transfers: Block Write and Block Read.

Block Write: WE_O='1', BLEN_O=3, BADDR_O=1234.



Block Read: WE_O='0', BLEN_O=3, BADDR_O=321.


CLK
CYC_O
VALID_I
READY_O
DAT_I
321
322
323
324