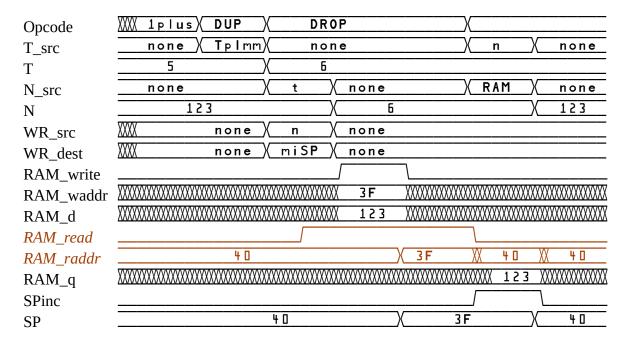
Block RAM

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:

Opcode	XXX 1plus X SWAP	(plus)	DUP	X		
T_src	none Tplmm	<u>n</u>)	Tpn	none		
T	5	<u></u>	123	129		
N_src	none	<u>t</u> \	RAM	<u>(t)</u>	none	
N	123	X	6	(321)	129	
WR_src	www.none			(<u>n</u>	none	
WR_dest	www.none			miSP X	none	
RAM_read			L			
RAM_raddr		XXXXXXX SP	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	(XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
RAM_q		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXX 321	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
SPinc						
SP		H D		\ <u> </u>		X 4 0
RAM_write						
RAM_waddr		XXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	(XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	X 4 D	XXXXXXXXXXXXXX
RAM_d	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	(XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	321	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

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.

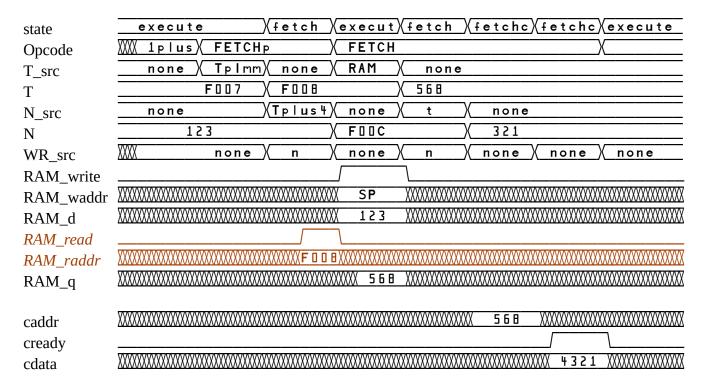


M32 Waveforms Page 1/2

Some opcodes request read and write operations. For example, >R.

Opcode	XXX toR X
T_src	none n none
T	5 321
N_src	none X RAM X none
N	321 \(123
WR_src	mone t none
_	
RAM_write	
RAM_write	\(\text{\lambda}\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
RAM_write RAM_waddr	
RAM_write RAM_waddr RAM_d	

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



M32 Waveforms Page 2/2