

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ Οργάνωση Υπολογιστών

Ομάδα: LAB31235453

Μπαλαμπάνης Ηλίας 2014030127 Μποκαλίδης Αναστάσιος 2014030069

Αναφορά Εργαστηρίου 3

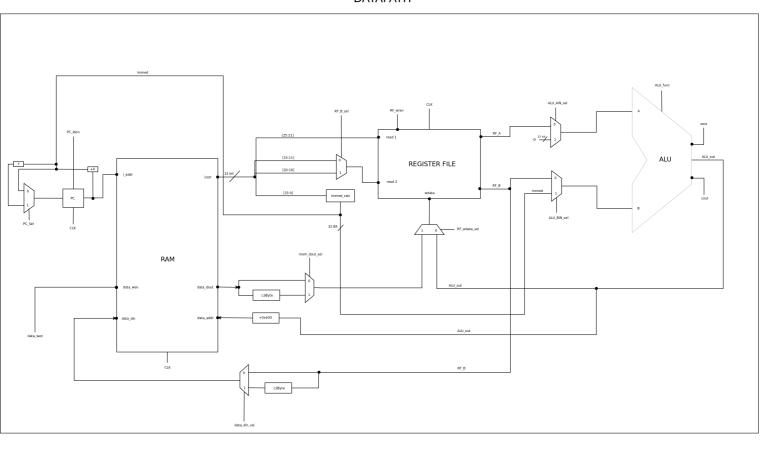
Σημείωση: Όλες οι εικόνες παρατίθενται με το .zip αρχείο της αναφοράς

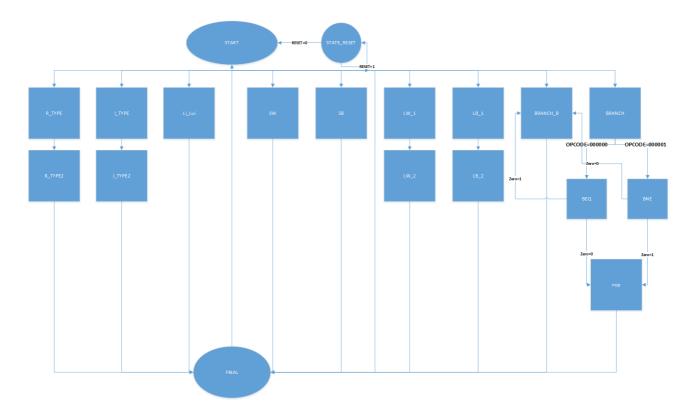
Περιγραφή Άσκησης

Μελετήσαμε όλα τα modules που υλοποίησαμε στα δύο προηγούμενα εργαστήρια και τα συνδέσαμε κατάλληλα έτσι ώστε να κατασκευάσουμε ένα ενιαίο datapath το οποίο φαίνεται παρακάτω.

To datapath σε **block diagram**:

DATAPATH





Η fsm μας ξεκινά από την κατάσταση state_reset στην οποία όσο δίνουμε reset=1 μένουμε σε αυτήν την κενή κατάσταση και ενεργοποιείται το reset του PC, δηλαδή στην έξοδο του έχουμε μηδέν. Με το που δώσουμε reset=0 τότε μεταβαίνουμε στη κατάσταση START στην οποία διαβάζουμε τον opcode και το function από τις εντολές που μας έρχονται από την μνήμη και αναλόγως επιλέγουμε σε ποια κατάσταση-πράξη θα μεταβούμε.

- **R_Type**: Σε αυτήν την κατάσταση αρχικά διαλέγουμε ποιό καταχωρητή θέλουμε να διαβάζουμε και σε ποιόν να γράψουμε δεδομένα. Επειδή η μονάδα των καταχωρητών, δηλαδή η RegisterFile, είναι σύγχρονη θα αναγκαστούμε να προσθέσουμε άλλη μια κατάσταση. Στην πρώτη διαβάζουμε τα δεδομένα από τους καταχωρητές και στην επόμενη κάνουμε την πράξη στην ALU και αποθηκεύουμε το αποτέλεσμα στον καταχωρητή που ορίζει το Instruction.
- ➤ **I_Type**: Αυτή η κατάσταση είναι παρόμοια με την προηγούμενη, απλώς η μόνη διαφορά είναι ότι αντί να διαβάζουμε ένα 2° καταχωρητή διαβάζουμε το immed που έχουμε. Αυτό το πετυχαίνουμε με την κατάλληλη ρύθμιση των σημάτων ελέγχου των πολυπλεκτών.
- Li_Lui: Αυτή η κατάσταση είναι παρόμοια με την I_type μονό που τώρα δεν διαβάζουμε από καταχωρητή αλλά παίρνουμε το immed το προσθέτουμε με τον μηδέν στην ALU και το αποτέλεσμα το αποθηκεύουμε στον επιθυμητό καταχωρητή. Έτσι στην ουσία γίνεται η αποθήκευση μιας σταθεράς σε ένα καταχωρητή. Επίσης άλλη μια διαφορά που υπάρχει με τις προηγούμενες καταστάσεις είναι ότι σε αυτήν την εντολή χρειαζόμαστε ένα κύκλο λιγότερο επειδή δεν διαβάζουμε από τους καταχωρητές.

- SW: Σε αυτήν την κατάσταση κάνουμε παρόμοια βήματα με την li_lui απλώς αντί να αποθηκεύουμε το αποτέλεσμα της ALU σε καταχωρητή την αποθηκεύουμε στην μνήμη. Σημαντικό σε αυτήν την περίπτωση είναι να αποθηκεύουμε στη σωστή θέση της μνήμης το αποτέλεσμα που θέλουμε. Αυτό το επιτυγχάνουμε με το να προσθέτουμε στην έξοδο της ALU το 0x400 και αυτό να πηγαίνει στην είσοδο ανάγνωσης διεύθυνσης της μνήμης RAM που φτιάξαμε.
- **SB**: Είναι η ίδια κατάσταση με την προηγούμενη μόνο που σε αυτήν την περίπτωση επιλέξουμε μέσω ενός καταχωρητή να αποθηκεύσουμε το αποτέλεσμα που θέλουμε σε bytes και όχι σε word.
- **LW**: Σε αυτήν την κατάσταση θέλουμε να φορτώσουμε σε ένα καταχωρητή μια τιμή από την μνήμη. Γι΄ αυτό χρειαζόμαστε 2 κύκλους. Ένα κύκλο όπου θα βγάλει την τιμή από την μνήμη και έναν που θα γράψει την τιμή στον επιθυμητό καταχωρητή. Επίσης στον 2° κύκλο θα αλλάξει και η διεύθυνση όπου θέλουμε να διαβάσουμε από την μνήμη έτσι ώστε να είναι έτοιμο να διαβάσει νέα δεδομένα σε αντίστοιχη εντολή.
- **LB**: Αυτή η κατάσταση είναι ίδια με την προηγούμενη μόνο που στην έξοδο της μνήμης επιλέγουμε τα δεδομένα που θα βγουν να είναι σε bytes και όχι σε word. Αυτό επιτυγχάνεται με το κατάλληλο σήμα ελέγχου ενός πολυπλέκτη.
- **Branch_B**: Σε αυτή τη κατάσταση στην ουσία αυξάνουμε τον PC σύμφωνα με το αποτέλεσμα της PC+4+Immed.
- Branch : Μέσω αυτής την κατάστασης ελέγχουμε πότε χρειαζόμαστε να κάνουμε BEQ ή BNE
- **BEQ**: Ελέγχουμε αν 2 καταχωρητές είναι ίσοι. Αυτό το υλοποιούμε αφαιρώντας τις τιμές στην ALU. Αν στην έξοδο της ALU έχουμε **Zero=1** δηλαδή οι καταχωρητές είναι ίσοι, εκτελούμε την εντολή **branch_b**. Διαφορετικά μεταβαίνουμε στην κατάσταση **nop** στην οποία απλώς αυξάνουμε τον PC κατά 4 έτσι ώστε να πάρουμε την νέα εντολή.
- **BNE**: Στην ουσία αυτή η εντολή είναι η αντίστροφη της BEQ. Αν έχουμε **Zero=0** εκτελούμε την εντολή **Branch_B**. Αντιθέτως μεταβαίνουμε στην κατάσταση **nop**.
- **Nop**: είναι μια κενή κατάσταση στην οποία ενεργοποιούμε το PC_LoadEnable και αυξάνουμε τον PC κατά 4.
- **FINAL**: είναι μια κενή κατάσταση στην οποία περιμένουμε να έρθει στον επόμενο κύκλο το νέο Instruction.

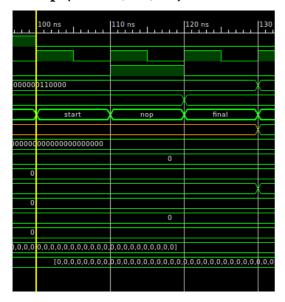
Κυματομορφές

Και με βάση το πρόγραμμα που μας δόθηκε έχουμε:

Reset για 10 κύκλους

Name	Value	110 ns	20 ns	30 ns	40 ns	150 ns	160 ns	170 ns	80 ns	90 ns	100 ns
	value	 			***************************************	201121111111111111111111111111111111111		70 ns		ـــــلــــــــــــــــــــــــــــــــ	لىسىلىتتت
le reset	0										
∏ _a clock	1										
ln pc_lden	0										
▶ 📑 instr[31:0]	100000000000000000000000000000000000000					100000000000000	00000000000011000	0			
▶ ₹ pc_out[31:0]	0						0				
16 current_state	start				stat	e_reset					start
▶ 📑 immed[31:0]	U						U				
▶ ■ alu_out[31:0]	000000000000000000000000000000000000000					000000000000000	000000000000000000000000000000000000000	0			
▶ 📑 ard1[4:0]	0						0				
▶ 📑 ard2[4:0]	0						0				
▶ 📑 awr[4:0]	0						0				
▶ 📑 din[31:0]	0						0				
▶ 📑 dout1[31:0]	0						0				
▶ 👫 dout2[31:0]	0						0				
▶ 🌃 sig_in_mux32[0:31]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0				[0,0,0,0]	,0,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0,0,0	[0,0,0,0,0,0			
▶ ■ ram[2047:0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0				0,0,0,0,0]	0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0			

Nop (add \$0, \$0, \$0)



li \$1, 2 -- \$1 = 2 li \$2, 3 -- \$2 = 3 li \$3, 4 -- \$3 = 4 li \$4, 5 -- \$4 = 5

/alue	ليتنين	140 ns	150 ns	160 ns	170 ns	180 ns	190 ns	200 ns	210 ns	220 ns	230 ns	240 ns
1000000000010000000000000000101	111000	00000000010000000	000000010	11100000	0000001000000000	0000011	11100000	0000001100000000	00000100	11100000	0000010000000000	0000101
		4		8			12			16		20
art	start	li_lui)	final	start	li_lui	final	start	li_lui	final	start	li_lui	final
		2			3			4			5	
	0	2			3			4			5	v •
0							0					
_	0	1			2			3			4	0
		1			2			3			4	
	0	2			3			4			5	0
							0					
	0	2			3			4			5	0
,2,3,4,0,0,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0	,0,0,0,0,0,0,0,0,0,0	[0,2,0,0,0,0,0,0,0,0,0,	0,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0	[0,2,3,0,0,0,0,0,0,0,	0,0,0,0,0,0,0,0,0,0,0,0,	0,0,0,0,0,0,0,0,0,0,0	[0,2,3,4,0,0,0,0,0,0,	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0	[0,2,3,4,5,0,0,0,
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					0,0,0,0,0]	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0				
16 ar	00000000120000000000121 t	00000000010000000000000000000000000000	11100000000010000000000000000000000000	11100000000010000000000000000000000000	111000000000100000010 11100000 4 8 start	11100000000001000000000000000000000000	111000000000010000000000010 11100000000	11100000000000100000000000000000000000	11100000000010000000000000000000000000	11100000000001000000000000000000000000	11100000000001000000000000000000000000	11100000000001000000000000000000000000

li \$5, -1 -- \$5 = -1

Name	Value		260 ns	270 ns
🗽 reset	0			
🗓 clock	1			
🏗 pc_iden	0			
▶ 📑 instr[31:0]	1000000010001100001000000110000	11100000	0000010111111111	1111111
pc_out[31:0]	24	2	p	
🖟 current_state	start	start	li_lui)	final
immed[31:0]	-1			
▶ 📆 alu_out[31:0]	6	0	-1	0
▶ 📑 ard1[4:0]	2		0	
▶ ■ ard2[4:0]	2	-1	5	-1
▶ 📑 awr[4:0]	6		5	
din[31:0]	6	0	-1	0
▶ 📑 dout1[31:0]	3		0	
▶ 📑 dout2[31:0]	3	0	-1	0
sig_in_mux32[0:31]	[0,2,3,4,5,-1,0,0,0,0,0,0,0,0,0,0,0,0,	[0,2,3,4,5,0,0,0,0,0	0,0,0,0,0,0,0,0,0,	[0,2,3,4,5
ram[2047:0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0			

Name	Value		270 ns	280 ns	290 ns	300 ns	310 ns	320 ns	330 ns	340 ns	350 ns
∏ _g reset	Θ										
🗓 clock	1										
l₀ pc_lden	0										
▶ 📑 instr[31:0]	1000000010001100001000000110000	11100000000000101	11111111111111		1000000001000110	0001000000110000		X	100000001100011	1001010000011000	4
pc_out[31:0]	24	20		2	4		*		2B		*
To current_state	start	li_lui	final	start	r_type	r_type2	final	start	r_type	r_type2	final
▶ 📑 immed[31:0]	-1						-1				
▶ 📆 alu_out[31:0]	6	-1	0					Х	7		5
▶ ■ ard1[4:0]	2	d			:	2		X		6	
▶ ■ ard2[4:0]	2	5	-1					X		5	
▶ ■ awr[4:0]	6					6		X		7	
▶ ■ din[31:0]	6	-1	0			6		X	7		5
▶ ■ dout1[31:0]	3	d						X		6	
▶ ■ dout2[31:0]	3	-1	0					X		-1	
sig_in_mux32[0:31]	[0,2,3,4,5,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,	[0,2,3,4,5,0,0,0,	[0,2,3,4,5	-1,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0	[0,2,3,4,5	,-1,6,0,0,0,0,0,0,0,0,0	,0,0,0,0,0,0,0,0,0,0	[0,0,0,0,0	[0,2,3,4,5,-1,6,7,0,0
ram[2047:0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,]		

Name	Value		350 ns	360 ns	370 ns	380 ns	390 ns	400 ns	410 ns	420 ns	430 ns
∏ _e reset	0										
/test_processor/reset	1										
l₀ pc_lden	0										
▶ ■ instr[31:0]	100000000110100000000000000111001	1000000011000111	.0010100000110		1000000001101000	0000000000111001		*	1100000010001001	000000000000000100	
▶ ■ pc_out[31:0]	32	28		3	12		*	3	6		*
1 current_state	start	r_type2	final	start	r_type	r_type2	final	start	i_type	i_type2	final
▶ 📑 immed[31:0]	-1			-3				*		4	
> = alu_out[31:0]	8	7	5		8		4	5		9	5
▶ ■ ard1[4:0]	3	€						*		4	
▶ ■ ard2[4:0]	0					0				9	*
▶ ■ awr[4:0]	8	7			1			*		9	
▶ ■ din[31:0]	8	7	5		8		4	5		9	5
▶ ■ dout1[31:0]	4	€						*		5	
▶ ■ dout2[31:0]	0	-1					0			9	*
▶ i sig_in_mux32[0:31]	[0,2,3,4,5,-1,6,7,0,0,0,0,0,0,0,0,0,0,	[0,2,3,4,5,-1,6,0	[0,2,3,4,5	-1,6,7,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0	0,0,0,0,0	[0,2,3,4,5	-1,6,7,8,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0]	[0,2,3,4,5,-1,6,7,8,9
▶ ■ ram[2047:0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					[0,0,0,0,0,0,0]	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0,0	,0,0,0,]		

Name	Value		450 ns	460 ns	470 ns	480 ns	490 ns	500 ns	510 ns	520 ns	530 ns	540 ns
om Out reset	0											
Llo clock	1											
l <mark>⊚</mark> pc_lden	0											
▶ 📑 instr[31:0]	10000000101010110000000000110100		10000000100010	0000000000011110	0	11100000	0000000000000000000	00001011	X	100000001010101	10000000000110100	
pc_out[31:0]	52		40			44			4	8		5:
turrent_state	final	start	r_type	r_type2	final	start	li_lui	final	start	r_type	r_type2	final
	11			4					11			
▶ ■ alu_out[31:0]	-1		10		5	0	11	0	X	0		-1
▶ 📑 ard1[4:0]	5			4			0		*		5	
▶ 📑 ard2[4:0]	Θ							0				
▶ 📑 awr[4:0]	11			10			0		X	1	11	
▶ 📑 din[31:0]	-1		10		5	0	11	0	X	0		-1
▶ ■ dout1[31:0]	-1			5			0		X		-1	
▶ 📑 dout2[31:0]	Θ							0				
sig_in_mux32[0:31]	[0,2,3,4,5,-1,6,7,8,9,10,0,0,0,0,0	[0,2,3,4,5,-1,6,	7,8,9,0,0,0,0,0,0,0,0,	0,0,0,0,0,0,0,0,0,			I	0,2,3,4,5,-1,6,7,8,9,1	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0]	
▶ 🖟 ram[2047:0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					0,0,0,0,0]	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0	.0,0,0,0,0,0,0			
▶ ■ alu_func[3:0]	0000		1100		0000		0000		*	0100		0000

nandi \$12, \$5, 1 -- \$12 = -2 nand \$13, \$12, \$12 -- \$13 = 1

	Value		530 ns	540 ns	550 ns	560 ns	570 ns	580 ns	590 ns	600 ns	610 ns	620 ns
To reset	θ											
V _a clock 1	1											
lopc_iden 0	8											
▶ 📑 instr[31:0]	110010001010110000000000000000000000000	10000000	10101011000000000	0110100		1100100010101100	000000000000000000000000000000000000000		*	100000011000110	0110000000110010	
▶ ■ pc_out[31:0] 5	52	4	3		5	2		k	:	56		60
to current_state	start	r_type	r_type2	final	start	i_type	i_type2	final	start	r_type	r_type2	final
▶ 📑 immed[31:0] 1	1		11						1			
▶ ■ alu_out[31:0] -	-1	C		-1	-1	-	2	1	*	1		4
▶ 📑 ard1[4:0] 5	5				5				(12	
▶ 📑 ard2[4:0] 0	0		C			1	2	0	.		12	
▶ 📑 awr[4:0]	12		11			1	2		*		13	
▶ 📑 din(31:0) -	-1	0		-1	-1		2	-1	*	1		4
▶ 📑 dout1[31:0] -	-1				-1				*		.2	
▶ 📑 dout2[31:0] 0	0			0			-2	0	*		.2	
▶ 🥌 sig_in_mux32[0:31]	[0,2,3,4,5,-1,6,7,8,9,10,0,0,0,0,0		[0,2,3,4,5,-	1,6,7,8,9,10,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0,0,0,	,0,0,0,0,0		[0,2,3,4,5,-	1,6,7,8,9,10,0,-2,0,0,	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0	[0,2,3,4,5,-1,6,7,8,9
▶ 🕌 ram[2047:0] [[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					,0,0,0,0,0,0,0	,0,0,0,0,0,0,0,0,0,0,0	,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,]			
▶ ■ alu_func[3:0]	9919	01	00	0000		0010		0000	*	0010		0000
_												

or \$14, \$13, \$1 -- \$14 = 3 sra \$15, \$8 -- \$15 = 4

Name	Value		620 ns	630 ns	640 ns	650 ns	660 ns	670 ns	680 ns	690 ns	700 ns
le reset	0										
To clock	1										
☐ pc_lden	Θ										
▶ 📑 instr[31:0]	10000001101011100000100000110011	1000000110001101	0110000000110		1000000110101110	0000100000110011			1000000100001111	0000000000111000	
0 1	60	56		6	0		K	6	4		X
10	start	r_type2	final	start	r_type	r_type2	final	start	r_type	r_type2	final
▶ 📑 immed[31:0]	1						1				
▶ ■ alu_out[31:0]	3	1	-4						4		8
	13	1	2		1	3				8	
▶ 📲 ard2[4:0]	1	1	2		1					0	
- 1 - m, m,	14	1	3		1	4			1	15	
▶ 📑 din[31:0]	3	1	-4						4		8
▶ 👫 dout1[31:0]	1	-2			1					8	
▶ 📑 dout2[31:0]	2	-2			7					0	
106	[0,2,3,4,5,-1,6,7,8,9,10,0,-2,1,0,		[0,2,3,4,5,-	1,6,7,8,9,10,0,-2,1,0,	0,0,0,0,0,0,0,0,0,0,0,0,0			1,6,7,8,9,10,0,-2,1,3,0		0,0,0,0,0,0]	[0,2,3,4,5,-1,6,7,8,9
- M	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					,0,0,0,0,0,0,0		0,0,0,0,0,0,0,0,0,0,0,0,0			
▶ 📑 alu_func[3:0]	0011	0010	0000		0011		0000		1000		0000
4											

ror \$16, \$10 -- \$16 = 5 ori \$17, \$4, 2 -- \$17 = 7, PC = 72

Name		Value		1700 ns	710 ns	1720 ns	1730 ns	1740 ns	1750 ns	1760 ps	1770 ps	1780 ns 15
la reset		value		700113	710113	720113	730113	740113	730113	700 113	770113	780 ns
		-	=									
1 clock		1										
la pc_lden		0										
instr[31:0]		100000010101000000000000000111101	1000000100001111	.0000000000111		1000000101010000	0000000000111101		<u> </u>	1100110010010001	000000000000000000000000000000000000000	 X
pc_out[31:0]		68	64		6	В		X	7	/2		ХТ
Current_state		start	r_type2	final	start	r_type	r_type2	final	start	i_type	i_type2	final
immed[31:0]	1	1			1						2	X
alu_out[31:0)]	5	4	8		5		10	5	X	7	5 1
ard1[4:0]		10				1	D				4	X
ard2[4:0]		0				0				-1	15	*
awr[4:0]		-16	1	5		-1	6			-1	15	 x
din[31:0]		5	4	8		5		10	5	*		5 **
dout1[31:0]		10				1	D				5	X
dout2[31:0]		0				C					7	x •
sig_in_mux3:	2[0:31]	[0,2,3,4,5,-1,6,7,8,9,10,0,-2,1,3,	[0,2,3,4,5,-1,6,7	[0,2,3,4,5,-	1,6,7,8,9,10,0,-2,1,3,	,0,0,0,0,0,0,0,0,0,0,0,0,0	,0,0,0,0,0,0	[0,2,3,4,5,-	1,6,7,8,9,10,0,-2,1,3,	4,5,0,0,0,0,0,0,0,0,0,	[0,0,0,0,0,0	[0,2,3,4,5,-1,6,7,8,9]
ram[2047:0]		[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					[0,0,0,0,0,0]	,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,]		
alu_func[3:0]	1	1101	1000	0000		1101		0000		0011		0000
				·	·				·		·	

Name	Value		780 ns	790 ns	800 ns	810 ns	820 ns	830 ns	840 ns
∏ _e reset	0								
∏ _a clock	1								
୍ଲା pc_lden	0								
▶ 📑 instr[31:0]	1110010000010100000000000000000001	1100110010010001		11100100	00010100000000000	00000001	11111100	00000000000000000	00000011
▶ ■ pc_out[31:0]	76	72		76			80		×
🖟 current_state	start	i_type2	final	start	(li_lui	final	start	branch_b	final
▶ 📷 immed[31:0]	65536	2			65536				
▶ 📑 alu_out[31:0]	0	7	5	(0	65536		(
▶ ■ ard1[4:0]	0	4				(
▶ ■ ard2[4:0]	0	-15			-12			0	
▶ 🕎 awr[4:0]	-12	-1	5		-12			0	
▶ 📑 din[31:0]	0	7	5	0	65536		(
▶ 📑 dout1[31:0]	0	5				(
▶ ■ dout2[31:0]	0	7			65536			0	
sig_in_mux32[0:31]	[0,2,3,4,5,-1,6,7,8,9,10,0,-2,1,3,	[0,2,3,4,5,-1,6,7	[0,2,3,4,5,-1,6,7,8,9	,10,0,-2,1,3,4,5,7,0,0	,0,0,0,0,0,0,0,0,0,0			[0,2,3,4,5,-1,6,7	8,9,10,0,-2,1,3,4,5,7,
▶ ■ ram[2047:0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					,0,0,0,0,0,0]	,0,0,0,0,0,0,0,0,0,0,0,0	,0,0,0,0,0,0,0,0,0,0,0,	0,0,0,0,]
alu_func[3:0]	0000	0011	0000				0000		

beq \$14, \$2, 3 -- PC = PC + 4 + 12 = 112 bne \$14, \$2, 16 -- PC = 116

	Malas	4										Acres 1
Name	Value		850 ns	860 ns	870 ns	880 ns	890 ns	900 ns	910 ns	920 ns	930 ns	940 ns
le reset	0											
∏ _e clock	1											
l pc_lden	0											
▶ 📑 instr[31:0]	00000001110000100000000000000011	111111000000		0000000	0111000010000000000	00000011			0000010	0111000010000000000	00001111	
pc_out[31:0]	96			96					112			116
To current_state	branch	final	start	branch	beq	branch_b	final	start	branch	bne	nop	final
▶ 🙀 immed[31:0]	12				AP					60		
▶ ■ alu_out[31:0]	0	0	3				3				<u> </u>	
▶ ■ ard1[4:0]	14	0						4				
▶ ■ ard2[4:0]	2		e e e e e e e e e e e e e e e e e e e		A STATE OF THE STA		0					0
▶ ■ awr[4:0]	2	0										
▶ ■ din[31:0]	0	0	3				3					
▶ 📑 dout1[31:0]	3	0										
▶ ■ dout2[31:0]	3		4				0					0
▶ ■ sig_in_mux32[0:31]	[0,2,3,4,5,-1,6,7,8,9,10,0,-2,1,3,					[0,2,3,4,5,-1,6,	.7,8,9,10,0,-2,1,3,4,5,7	0,0,65536,0,0,0,0,0	.0,0,0,0,0,0]			
	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					0.0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0	0,0,0,0,]			
▶ ■ alu_func[3:0]	0001	0	000	9	0001		0000		0	0001		0000
												4
											-	

sw \$5, 4(\$3) -- MEM[1026] = -1 sb \$10, 8(\$15) -- MEM[1027] = 10 lw \$18, 2(\$10) -- \$18 = MEM[1027] = 10

Nam		Value		1,000 ns	1,010 ns	1,020 ns	1,030 ns	1,040 ns	1,050 ns	1,060 ns	1,070 ns	1,080 ns	1,090 ns
	reset	Θ											
	clock	1											
1,	pc_lden	0											
▶	instr[31:0]	001111010101001000000000000000000000000	000000011	01111100	0110010100000000	00000100	00011101	1110101000000000	00001000		001111010101010010	000000000000000000000000000000000000000	
>	pc_out[31:0]	132		120		k	124			1	28		13
	current_state	final	final	start	SW	final	start	sb	final	start	lw_1	lw_2	final
ă	immed[31:0]	2	60		4			8					
i i	alu_out[31:0]	10		4	8			12	4	10	1	.2	10
ě	ard1[4:0]	10	15		3			15			1	.D	
¥	ard2[4:0]	0		0	5	*		10			-1	14	(
	awr[4:0]	-14	10		5			10			-1	14	
■	din(31:0)	10		4	8	K		12	4	10	-1	10	10
•	dout1[31:0]	10				4					1	.D	
	dout2[31:0]	0		0	-1	*		10		0		10	
*	sig_in_mux32[0:31]	[0,2,3,4,5,-1,6,7,8,9,10,0,-2,1,3,				[0,2,3,4,5,-	,6,7,8,9,10,0,-2,1,3,4	,5,7,0,0,65536,0,0,0	0,0,0,0,0,0,0]				[0,2,3,4,5,-1,6,7,8,9
•	ram[2047:0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0					0,0,0,0,0]	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,]			
•	alu_func[3:0]	0000							0000				

			000000000000000000000000000000000000000
	[1030]		000000000000000000000000000000000000000
ightharpoons	(1029)	000000000000000000000000000000000000000	00000000000000000000000000000000000000
\triangleright	(1028)	000000000000000000000000000000000000000	000000000000000000000000000000000000000
₽	[1027]	10	10
Þ	[1026]	-1	-1
ightharpoons	[1025]	000000000000000000000000000000000000000	000000000000000000000000000000000000000
ightharpoons	[1024]	000000000000000000000000000000000000000	000000000000000000000000000000000000000
₽	(1023)	000000000000000000000000000000000000000	000000000000000000000000000000000000000
h	110221	0000000000000000000000000000	000000000000000000000000000000000000000

lb \$19, 8(\$0) -- \$19 = MEM[1026] = 255

Name		Value	46.7	11 100 ps	11 110 ps	120 ps	11 130 ns	11 140 ns
		Value		1,100 lis	1,110 ns 1	,120 hs	1,130 lis	1,140 115
To reset		0						
V_{lpha} clock		1						
l₀ pc_lden		1						
instr[31:0]		0000110000010011000000000000000000	00		00001100000100110	0000000000001000		
pc_out[31:0]		132			132		136	
튢 current_state		1b_2	final	start		(Ib_2)	final	
▶ 📑 immed[31:0]		8	2		8			
alu_out[31:0]		8	10	× • • • • • • • • • • • • • • • • • • •	4		× • • •	
▶ ■ ard1[4:0]		0	10		4			
▶ 📑 ard2[4:0]		-13		0	-1		0	
▶ 📑 awr[4:0]		-13	-14		-1			
din[31:0]		255	10		A STATE OF THE STA	(255	0	
▶ 📑 dout1[31:0]		0	10		•			
▶ 📑 dout2[31:0]		255		0		(255	0	
sig_in_mux32[0:3	31]	[0,2,3,4,5,-1,6,7,8,9,10,0,-2,1,3,	[0,2,3	4,5,-1,6,7,8,9,10,0,-2	1,3,4,5,7,10,0,65536	0,0,0,0,0,0,0,0,0,	[0,2,3,4,5,-1,6,7,	
▶ 🎆 ram[2047:0]		[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0		[0,0,0,0,0,0,0,0]	,0,0,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0,0,0	0,0,0,]	
alu_func[3:0]		0000			0000			
		<i>(</i>						
i								