Rapport

CPU'en (8-bit) bruker 4 maskinkode-instruksjoner: laste inn registeret med 4 LSB / MSB, laste inn fra / skrive til dataminne.
Instruksjonen blir dekodert ved hjelp av en indre krets «instruksjonsdekoder». Den får de 4 MSB fra programminnet via splitter og velger hvilke utganger som skal være høye (vha. controlled buffer / tri-state. Trengs pga bus).
Jeg laget en sannhetstabell for å finne funksjonsuttrykkene til dekoderen.

(i)	(ii)	(iii)	(iv)	(v)	(vi)	
0000	0	0	0	0	10	
0001	0	0	0	0	10	
0010	0	0	0	0	10	
0011	0	0	0	0	10	
0100	0	0	0	0	10	
0101	0	0	0	0	10	
0110	0	0	0	0	10	
0111	0	0	0	0	10	
1000	0	1	0	0	00	
1001	0	1	0	0	01	4 instruksjoner
1010	0	0	1	0	11	
1011	1	0	0	1	10	
1100	0	0	0	0	10	
1101	0	0	0	0	10	
1110	0	0	0	0	10	
1111	0	0	0	0	10	

Hvorav:
(i) instruksjon (abcd)
(ii) RAM store
(iii) programminne enable
(iv) dataminne enable
(v) R enable
(vi) Reg.-select

Funksjonsuttrykkene:
(ii) ab'cd
(iii) ab'c'd' + ab'c'd
(iv) ab'cd'
(v) ab'cd
(vi) I. a' + b + c
II. ab'c'd + ab'cd'

Reg.-selector:
2-bit instruk.
00 -> load 4 LSB
01 -> load 4 MSB
10 -> behold verdien
11 -> load fra RAM
Realisert vha multiplexer.

Programtelleren (PC) er konstruert slik at den stopper ved 15 (0 til 15 = 16). Realisert vha OR-port – dens utgang (alltid 1 med mindre inngangen er 0000) er inngangen til PC sin enable.

Jeg brukte logisim sin default-adder for å telle antall kjørte instruksjoner. Tror, jeg har vist at jeg kan lage addere i forrige oblig. :)

Resten burde være pretty straight forward og lett lesbar.
Fått mail fra Viet Thi Tran om at jeg ikke trenger å logge kretsen.
«Logging.... du kan se bort fra dette. Det brukes for å debuge eller få
bedre forståelse hva gjør kretser.
Så lenge din oblig fungerer skal du lett få godkjent.»