Terminology

state $\rightarrow$ is a particular configuration | coding of storage (memory and registers), and a state transition is a change of the configuration | coding;

cycle $\rightarrow$ is the time between state transitions. If the storage is registers, we have an internal | machine | processor cycle, if the storage is memory, we have a memory cycle;

instruction $\rightarrow$ effects a state change;

computing machine $\rightarrow$ is defined by a set of instructions and some storage, together with an implementation, including a processor, registers, memory (and i/o), that realizes the state transitions defined by the instructions;

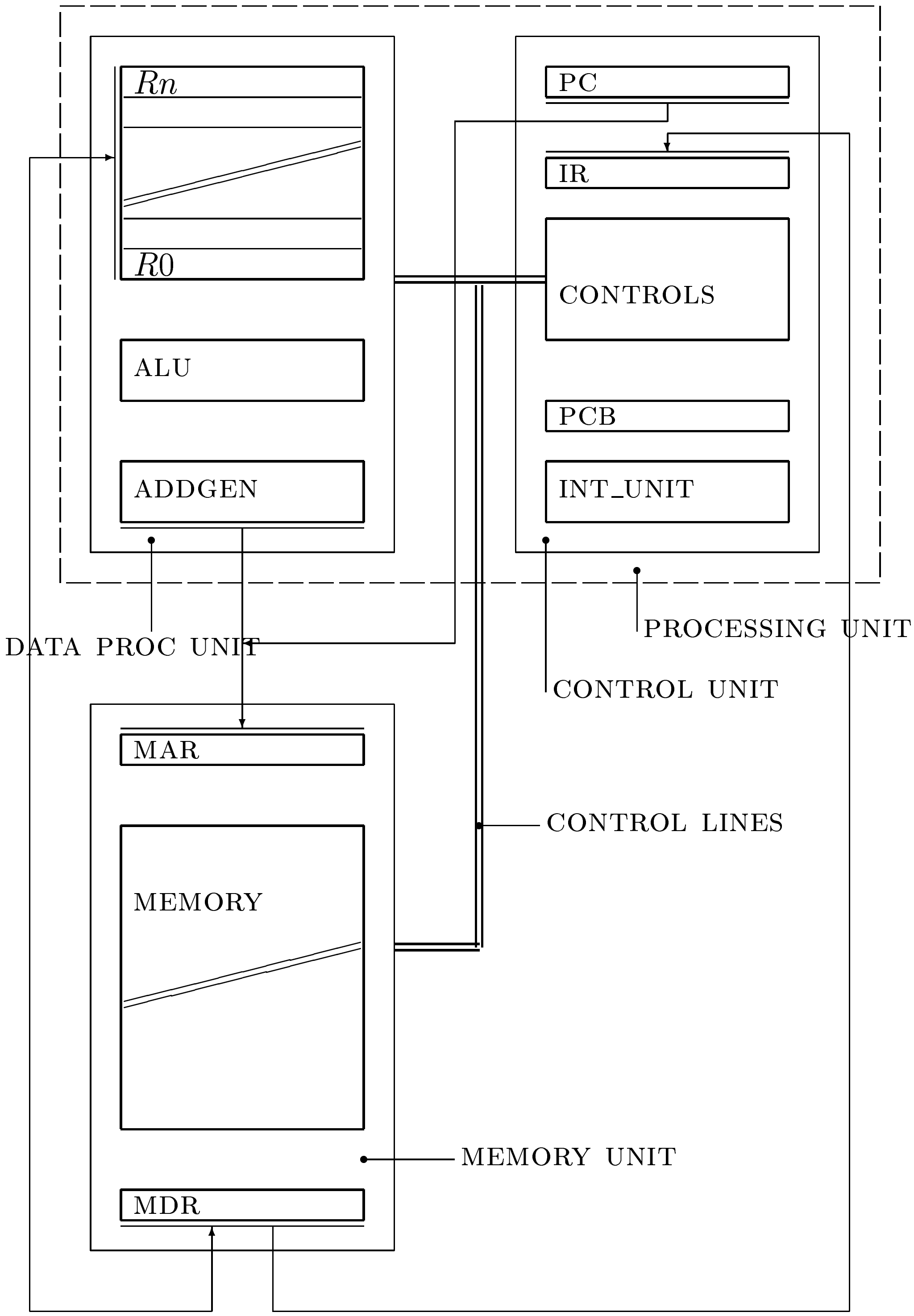
machine program $\rightarrow$ is a sequence of instructions;

process $\rightarrow$ is a program which transforms and initial into a final | terminal state;

storage $\rightarrow$ is the domain and range of the instructions. Hidden registers | memory that cannot be explicitely referred by instructions are not part of the storage but of the machine's implementation.

Instruction set mnemonics

class	mnemonic	operation
move instr	MOVE.DU	moves unpacked decimals mem_to_mem
	MOVE.C	moves character strings mem_to_mem
	PK.DU	packs decimals mem_to_mem
	UNPK.DP	unpacks packed decimals mem_to_mem
	MOVE.X	moves fixed formats reg_to_reg
	LD.X	loads fixed formats mem_to_reg
	ST.X	stores fixed formats reg_to_mem
	LDM.X	loads multiple fixed formats mem_to_reg
	STM.X	stores multiple fixed formats reg_to_mem
	LDA.W	loads address into reg
	SHL.U	shifts fixed formats left
	SHR.U	shifts fixed formats right
	CLR.X	clears fixed formats (registers)
arithmetic/ logic instr	ADD.X	addition
	SUB.X	subtraction
	MULT.X	multiplication
	DIV.X	division
	INC.U	increment
	DEC.U	decrement
	COMP.U	compares integers
	CMP.C	compares character strings
	CMP.DP	compares packed decimals
branch instr	BC.CC	(un-)conditional branch
	BR_DEC	branch on decrement (loop control)
	BSR	branch to subroutine
	RET	return from subroutine


Rn
$R0$
ALU
ADDGEN
PC
IR
CONTROLS
PCB
INT_UNIT
DATA PROC UNIT
PROCESSING UNIT
CONTROL UNIT
MAR
MEMORY
CONTROL LINES
MEMORY UNIT
MDR

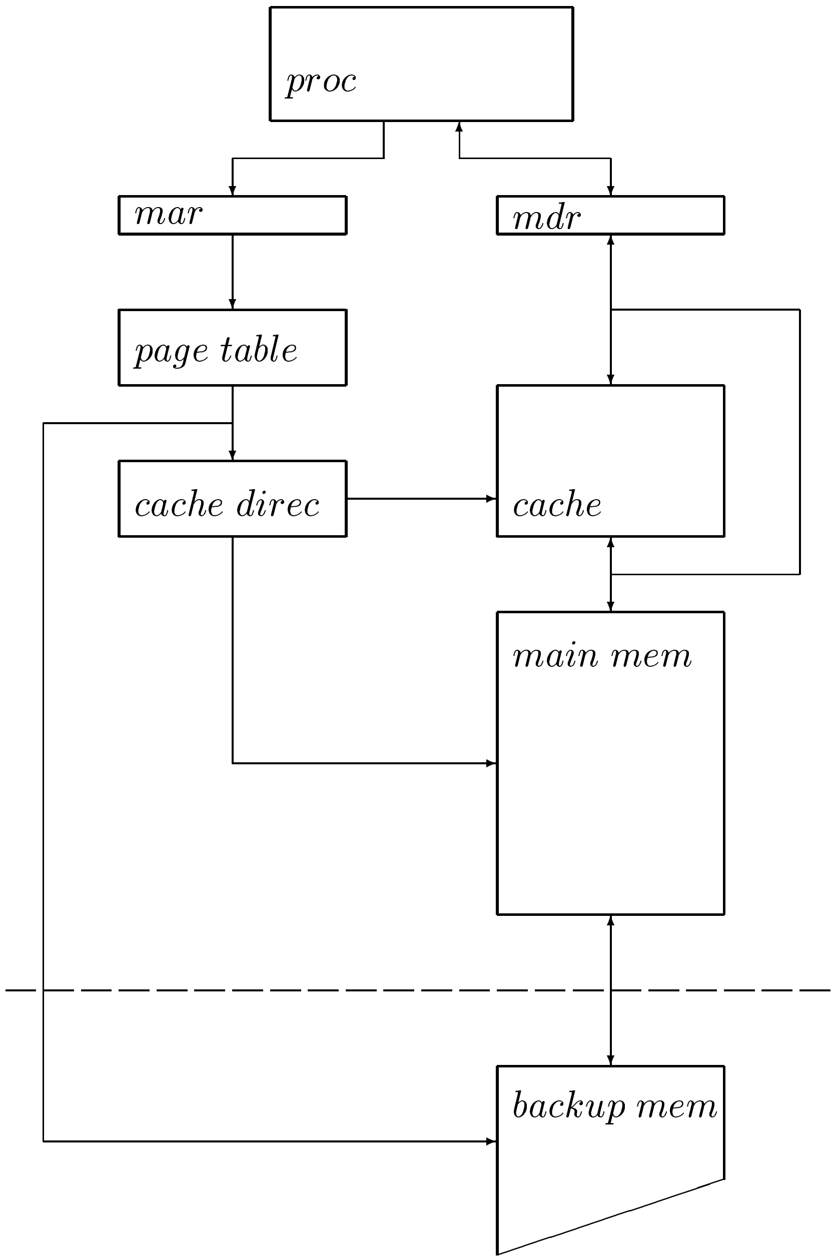
Meaning of the subscripts

modifier	type | format
B	integer byte | 8 bits
H	integer half word | 16 bits
W	integer (full) word | 16 bits
F	floating point full word | 16 bits
D	floating point double word | 64 bits
C	character string | $\leq$ 256 bytes
DD	decimal string unpacked | $\leq$ 16 bytes (digits)
P	decimal string packed | $\leq$ 16 bytes (digits)
X	B | H | W | F | D
U	B | H | W

Branch condition codes

CC mnemonic	meaning
`T`	$true \rightarrow$ branch always
`F`	$false \rightarrow$ branch never (go in-line)
`V`	arithmetic result overflow
`C`	arithmetic carry out
`N`	negative arithmetic result
`Z`	zero arithmetic result

Note that condition codes are
$\rightarrow$ set by arithmetic | logic instructions;
$\rightarrow$ inspected by subsequent `BC.CC` instructions in which `CC` may either be specified as `T` or `F` or as any combination of `V` | `C` | `N` | `Z`. as condition code.


proc
mar
mdr
page table
cache direc
cache
main mem
backup mem