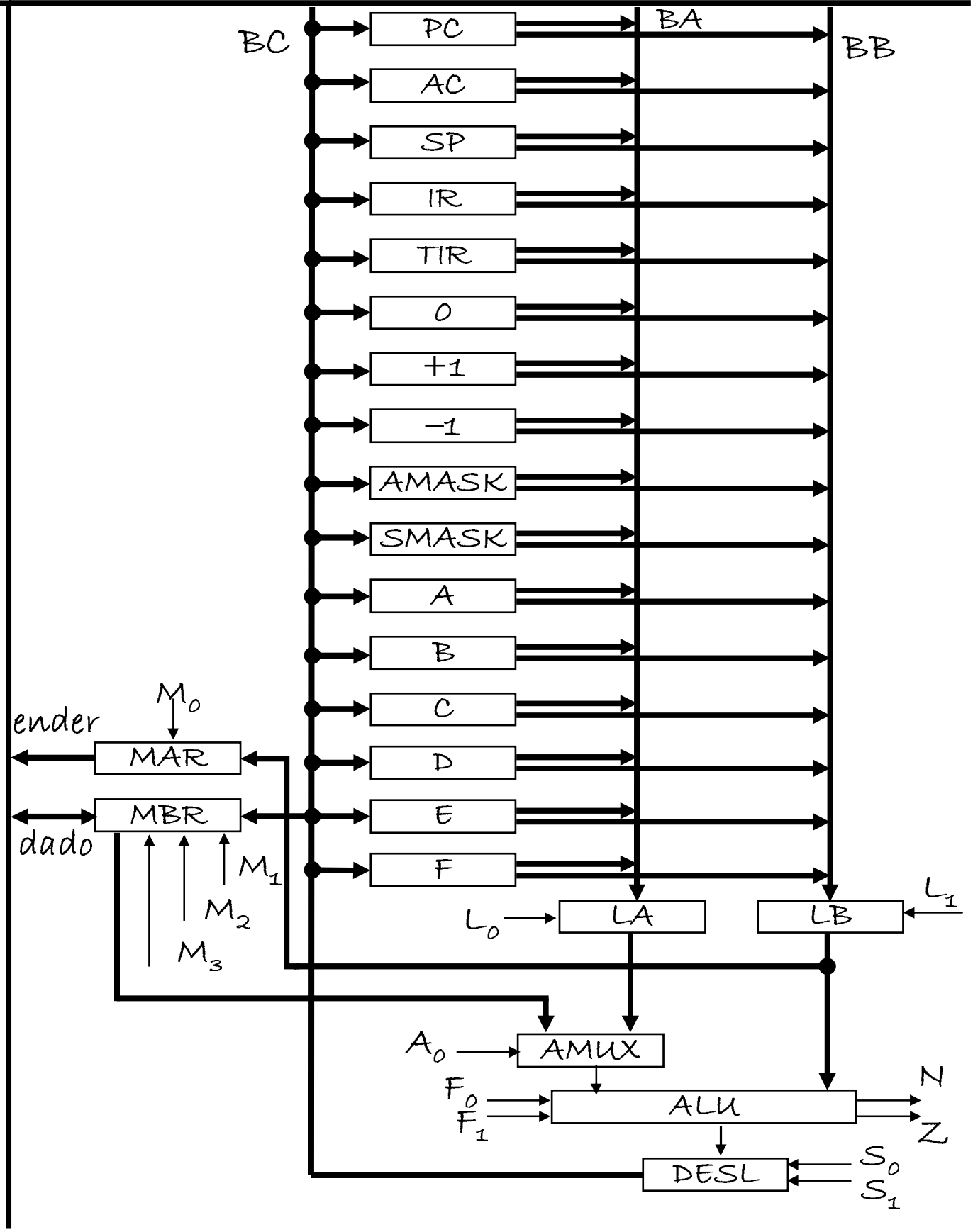
UNIDADE DE CONTROLE



UNIDADE DE CONTROLE

- 0000xxxxxxxxxxxx **LODD** AC := M[x]
- 0001xxxxxxxxxxxx **STOD** M[x] := AC
- 0010xxxxxxxxxxxx **ADDD** AC := AC + M[x]
- 0011xxxxxxxxxxxx **SUBD** AC := AC - M[x]
- 0100xxxxxxxxxxxx **JPOS** se AC > 0 então PC := x
- 0101xxxxxxxxxxxx **JZER** se AC = 0 então PC := x
- 0110xxxxxxxxxxxx **JUMP** PC := x
- 0111xxxxxxxxxxxx **LOCO** AC := x
- 1000xxxxxxxxxxxx **LODL** AC := M[SP + x]
- 1001xxxxxxxxxxxx **STOL** M[SP + x] := AC
- 1010xxxxxxxxxxxx **ADDL** AC := AC + M[SP+x]
- 1011xxxxxxxxxxxx **SUBL** AC := AC - M[SP+x]
- 1100xxxxxxxxxxxx **JNEG** se AC < 0 então PC := x
- 1101xxxxxxxxxxxx **JNZE** se AC /= 0 então PC := x
- 1110xxxxxxxxxxxx **CALL** SP := SP - 1; M[SP]:= PC;
- PC := x
- 1111000000000000 **PSHI** SP := SP - 1; M[SP] :=M[AC]
- 1111001000000000 **POPI** M[AC] := M[SP]; SP := SP + 1
- 1111010000000000 **PUSH** SP := SP - 1; M[SP] := AC
- 1111011000000000 **POP** AC := M[SP]; SP := SP + 1
- 1111100000000000 **RETN** PC := M[SP]; SP := SP + 1
- 1111101000000000 **SWAP** TMP := AC; AC := SP;SP := TMP
- 11111100yyyyyyyy **INSP** SP := SP + y
- 11111110yyyyyyyy **DESP** SP := SP - y

UNIDADE DE CONTROLE

• 0000	**ADD**	r1 := r1 + r2
• 0001	**AND**	r1 := r1 . r2
• 0010	**MOVE**	r1 := r2
• 0011	**COMPL**	r1 := inv (r2)
• 0100	**LSHIFT**	r1 := lshift (r2)
• 0101	**RSHIFT**	r1 := rshift (r2)
• 0110	**GETMBR**	r1 := MBR
• 0111	**TEST**	if r2 < 0 then N := true;
•		if r2 = 0 then Z := true
• 1000	**BEGRD**	MAR := r1; rd
• 1001	**BEGWR**	MAR := r1; MBR := r2; wr
• 1010	**CONRD**	RD
• 1011	**CONWR**	WR
• 1100		
• 1101	**NJUMP**	if N then goto r
• 1110	**ZJUMP**	if Z then goto r
• 1111	**UJUMP**	goto r