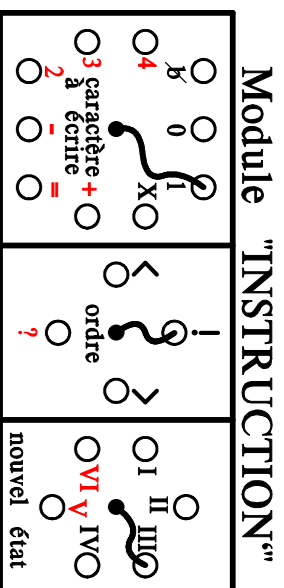
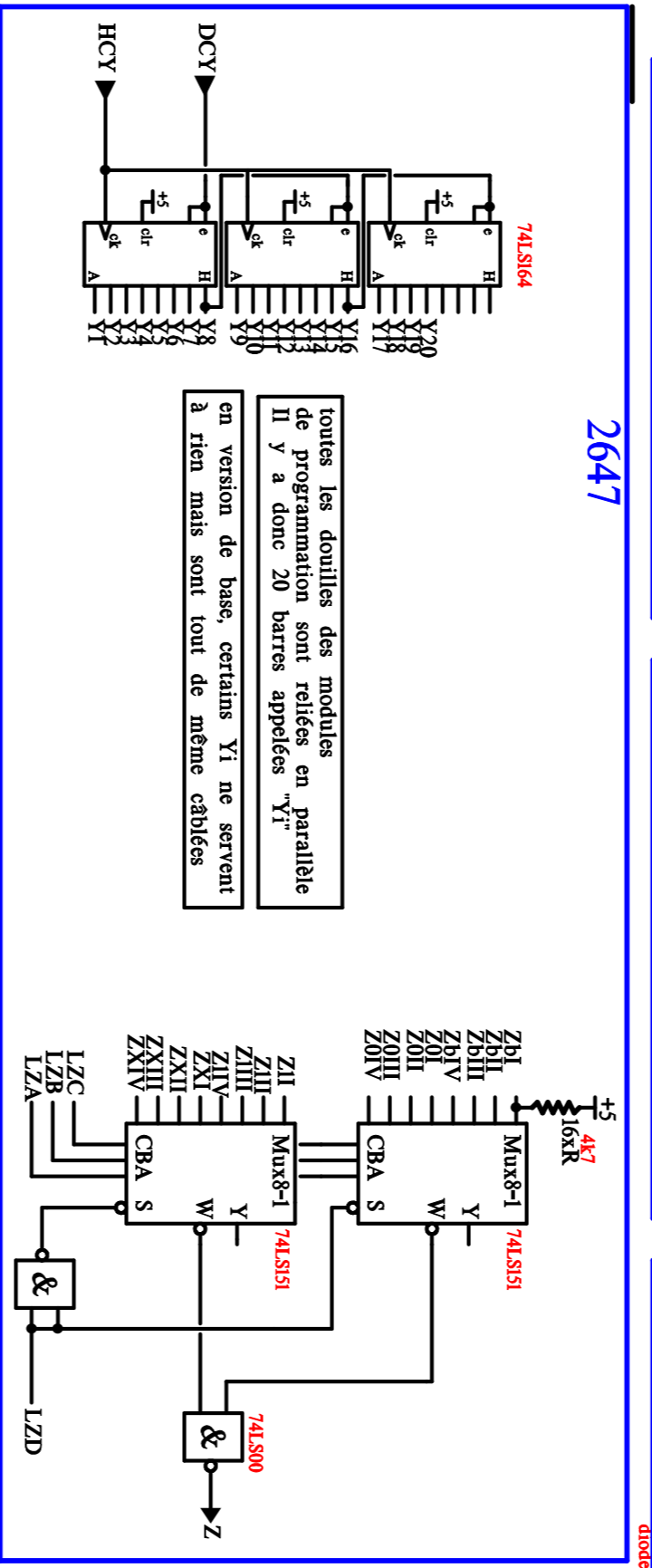
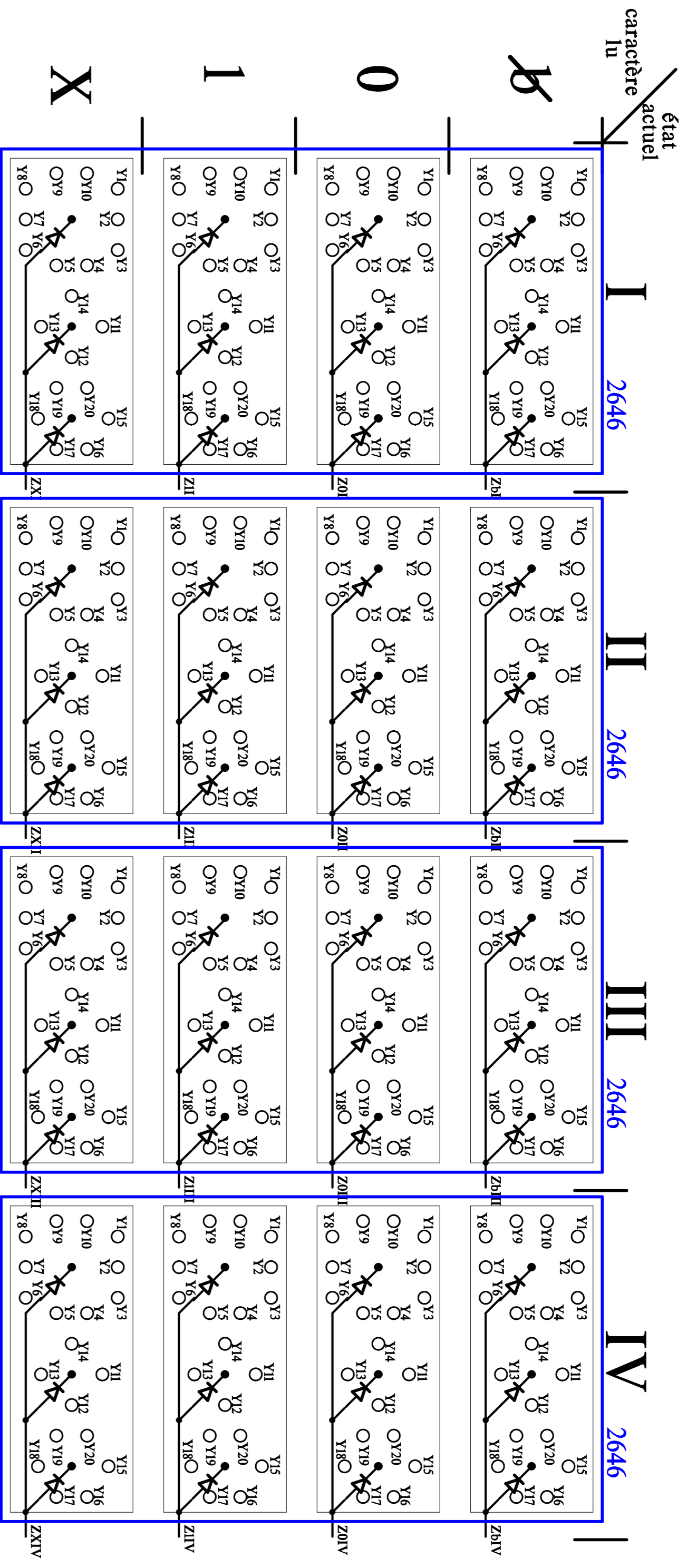


la programmation d'une instruction est réalisée par 3 straps :
 - caractère
 - ordre
 - état
 (par défaut, si le strap est absent, il n'y a pas de changement!)



version de base en noir :
 4 caractères
 3 ordres
 4 états
 extensions possibles en rouge:
 10 caractères (à définir)
 4 ordres (?=erreur)
 6 états



toutes les douilles des modules de programmation sont reliées en parallèle II y a donc 20 barres appelées 'Yi' en version de base, certains Yi ne servent à rien mais sont tout de même câblés

LZD	LZC	LZB	LZA	Z
0	0	0	0	ZbI
0	0	0	1	ZbII
0	0	1	0	ZbIII
0	0	1	1	ZbIV
0	1	0	0	ZoI
0	1	0	1	ZoII
0	1	1	0	ZoIII
0	1	1	1	ZoIV
1	0	0	0	ZI
1	0	0	1	ZII
1	0	1	0	ZIII
1	0	1	1	ZIIIV
1	1	0	0	ZXI
1	1	0	1	ZXII
1	1	1	0	ZXIII
1	1	1	1	ZXIIIV

Principe algorithme :
 commuter Z sur le module correspondant à l'état actuel et au caractère lu grâce au mot LZ sur DCY.HCY en le décalant 20 fois
 lire Z, en déduire :
 - le caractère à écrire (YI..Y10)
 - l'ordre (Y11..Y14)
 - le nouvel état (Y15..Y20)
 (la présence du strap de programmation est détectée par Z=0)

le mot LZ permet de commuter la sortie du module choisi sur la sortie Z, afin de lire l'instruction correspondante.
 extensions : il faut ajouter des circuits MUX8-1 et des bits de commande LZELZF, jusqu'à obtenir le nombre de modules instruction suivant le nombre d'états et de caractères de l'alphabet (6x10=60)

Mémoire morte de programme