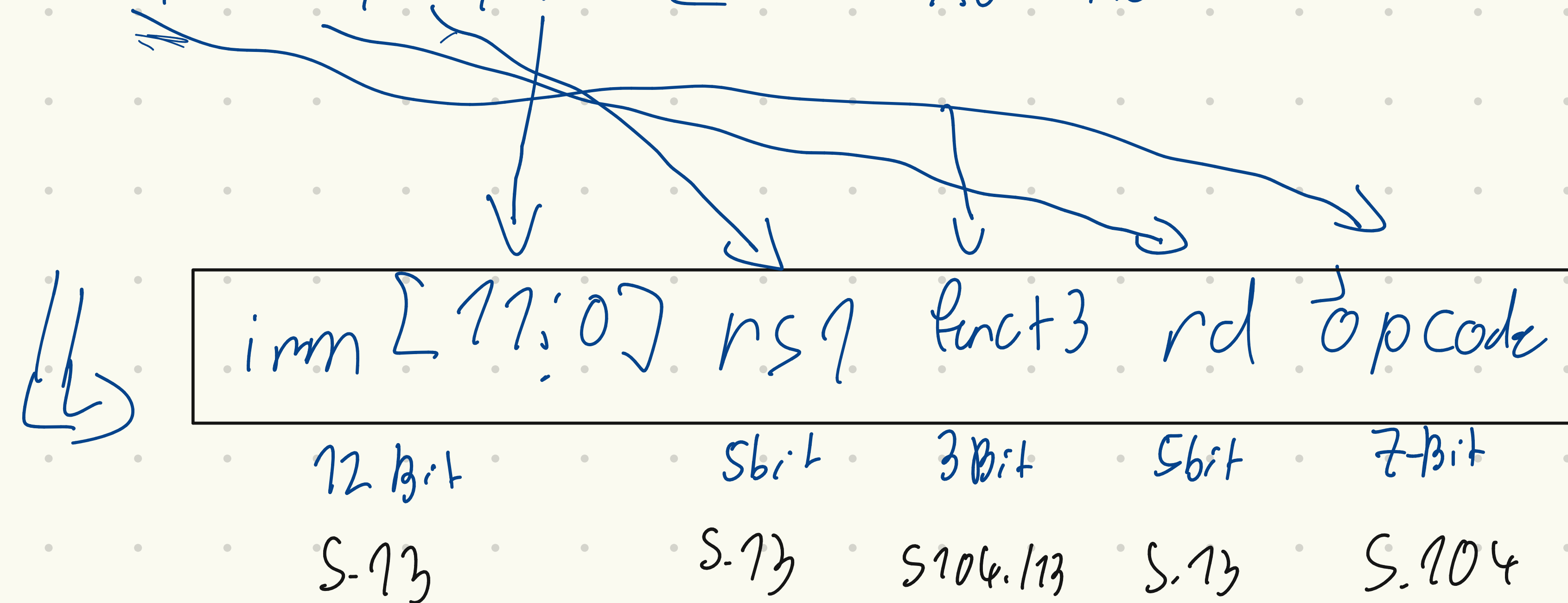


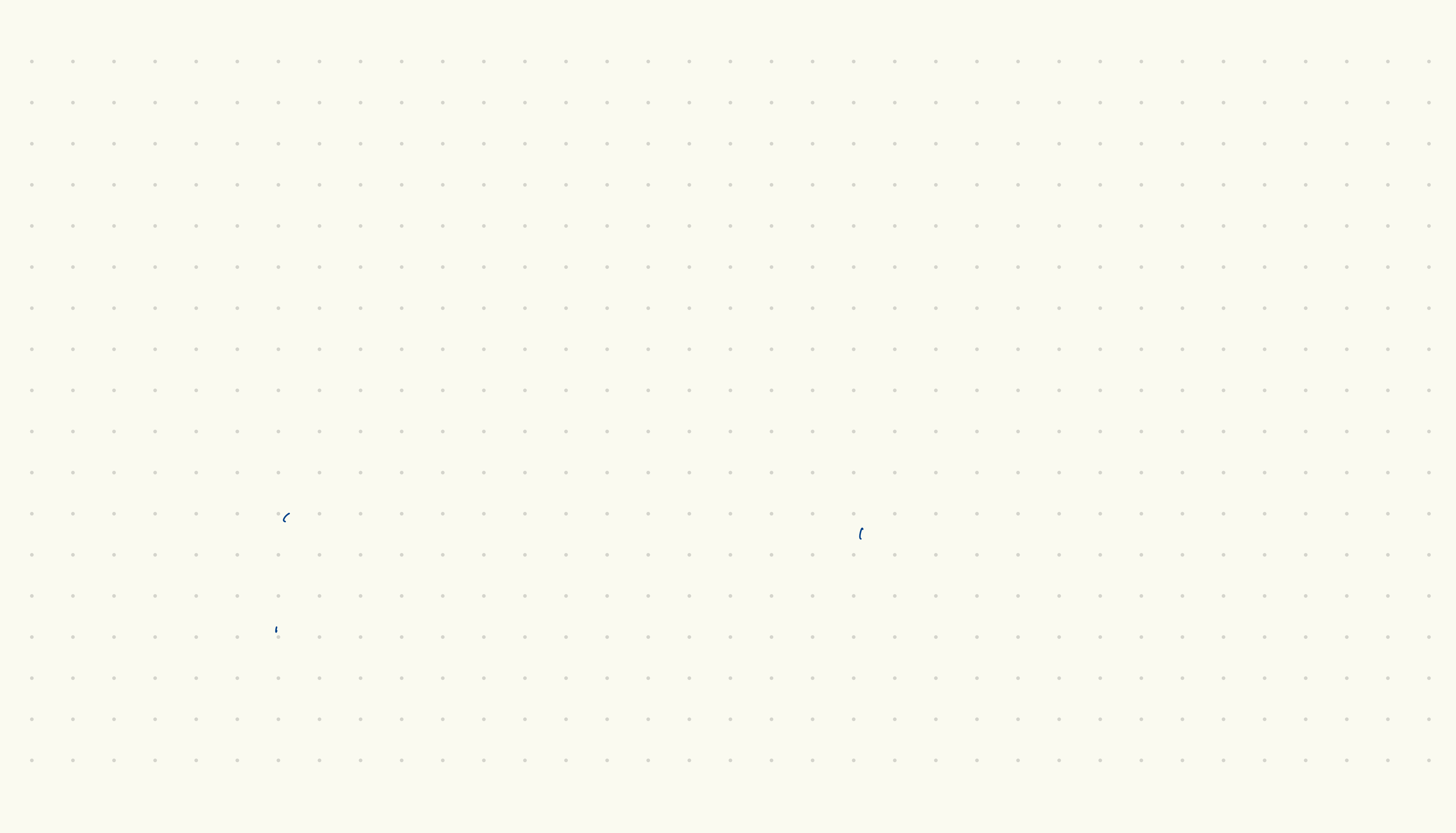
x0 Register in Manual book immer auf 0.

addi x6, x0, 7 \Leftrightarrow x6 = x0 + 7 S. 13



LL)

00 00000000111 00000 000 00110 0010011
7 x0 funct3 x6 addi



1.

addi x6, x0, 7 = 000000000111 00000 000 00110 0010011
7 x0 funct3 x6 addi

ori x2, x0, 5 = 00000000101 00000 000 00010 0010011
5 x0 funct3 x2 ori

add x31, x6, x2 = 00000000 00010 000 00110 11111
x2 funct3 x6 x31

2.

addi x6, x0, 7 \Rightarrow x6 = 7

ori x2, x0, 5 \Rightarrow x2 = 5

add x31, x6, x2 \Rightarrow x31 = x6 + x2 = 7 + 5 = 12

3.

Das Register PC hält die Adresse des nächsten auszuführenden Befehls.

Im Beispiel haben wir es benutzt indem wir ab Adresse 0x7000 die Befehle hintereinander ausgeführt haben.

Der Fetch-execute-Zyklus beschreibt den das der Prozessor die Befehle aufruft und ausführt.

jal x0, loop

1111111100111111 000000 1101111

jal x0, loop Jump and link

Jump and link springt zur Adresse wo das Label loop hinzeigt und speichert diese Adresse in das angegebene Register da aber x0 fest verdrahtet ist spielt das hier keine Rolle also werden hier die Register x6, x2 immer bei jedem Durchlauf mit +7 und +5 befüllt und dann wieder in x31 zusammen addiert.