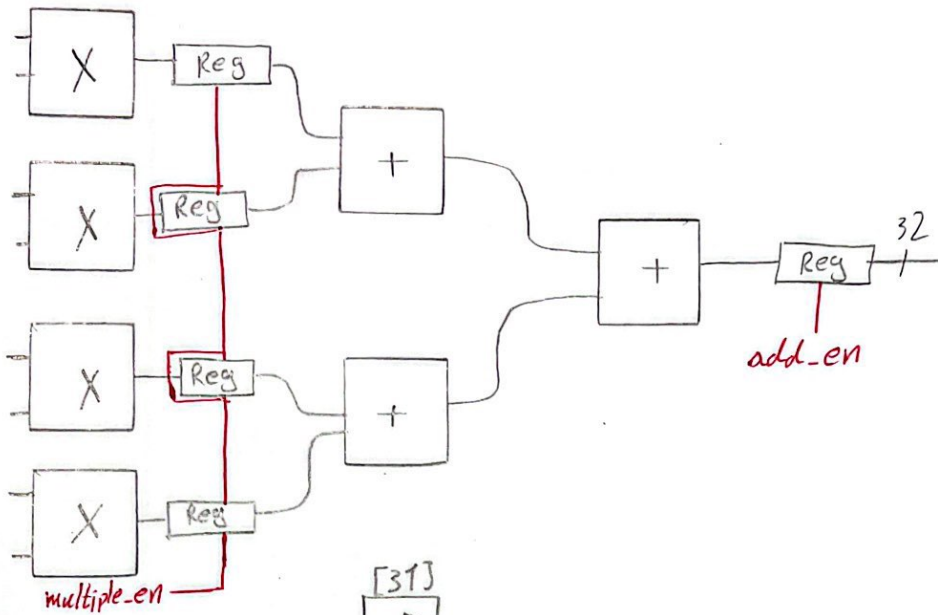
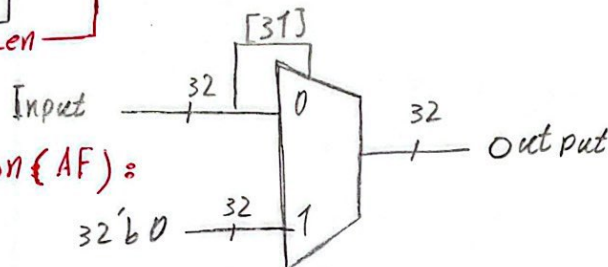


PU:



Activation function (AF):



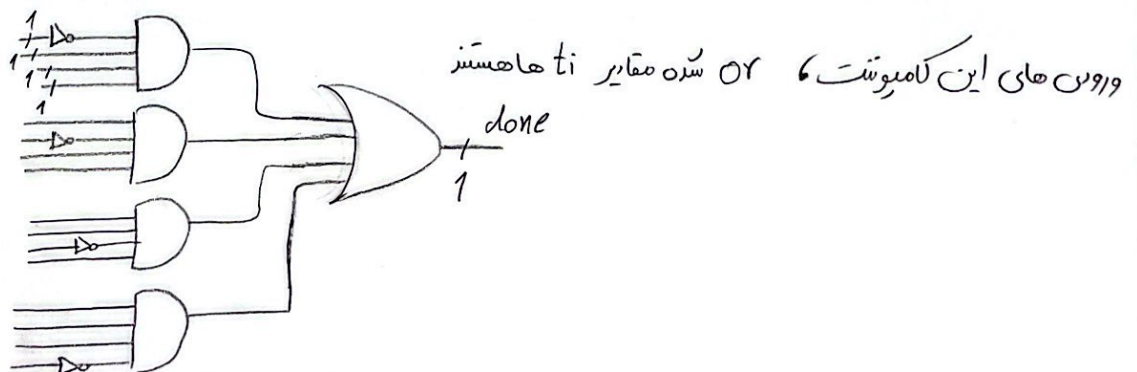
encoder:

ما یک encoder داریم که OR شده مقایره داخل رجیسترهای t_i را میگیرد و خروجی آن دو بیت خواهد بود که نشان می دهد کدام x جواب خوبی ما است. اگر همه t_i ها به جز t_1 صفر بودند، جواب ن خواهد بود. اگر همگی صفر باشند، خروجی آن 00 خواهد بود چرا که همه اعداد با هم برابرند.

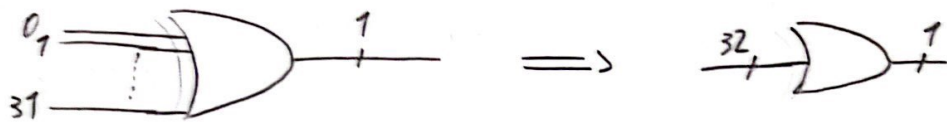
weights:

یک کامیونتی است که مقایره وزن ها را در خود دارد (buffer یا memory). برای راحتی از این کامیونتی به لایه های سیم رسم می کنیم.

done-checker:

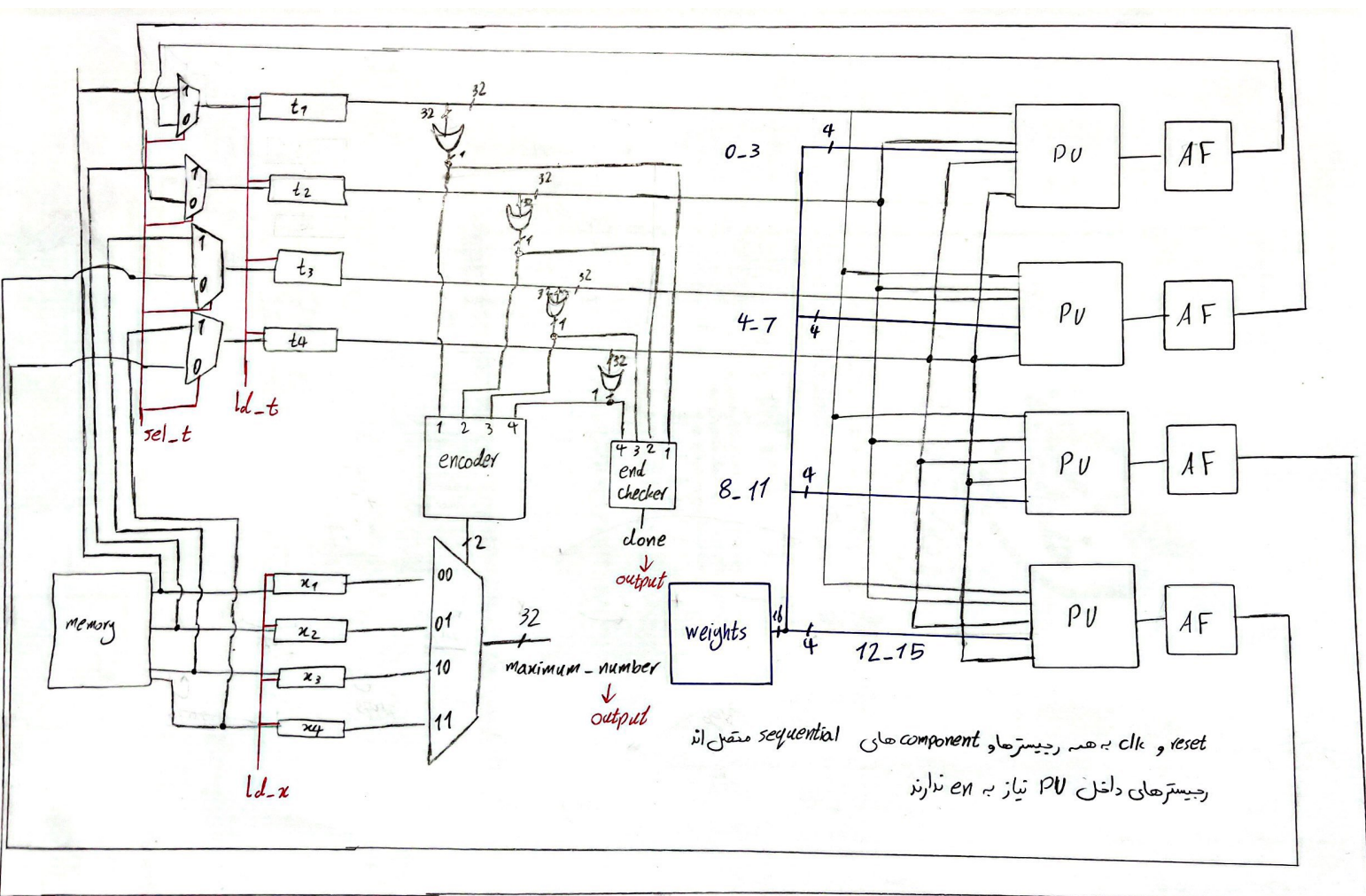


bitwise ORs



Memory:

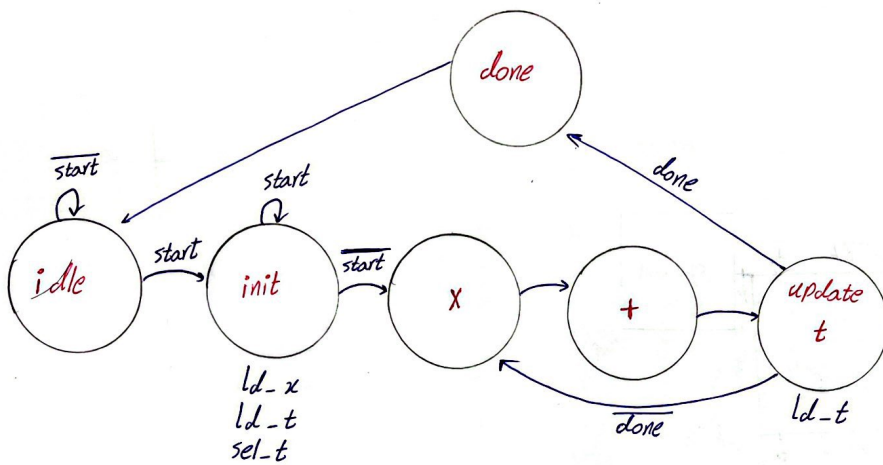
یک حافظه است که مقادیر اولیه پردازنده را در رجیسترهای x لود می‌کند



reset, clk به همه رجیسترها و component های sequential متصل اند
 رجیسترهای داخل PU نیاز به en ندارند

می‌شود به گونه‌ای تنظیم کرد که done خروجی controller باشد که به این منظور
می‌بایست خروجی زتاها به controller برود

محاسبه می‌شود پس از عملیات sum، خروجی را برسی کرد
که این کار در تمام برنامه ما را 1 گلاک اضافه می‌کند



(controller) خروجی ها {
ld-x
ld-t
sel-t

(controller) ورودی ها {
start
done

