ICLab Lab04 Exercise

Yu-Fan, Huang

2022 /10 /26

Outline

- **Pipeline Design**
  1. Data Flow
  2. Critical Issues: Timing and Area
  3. Stages
- **Reduce Register and Cycle Count**
  1. Reordering pipeline schedule
  2. Changing pipeline stages

Pipeline Design: Data Flow

- Data flow:

1. $(\mathrm{U} * X_1 + W * H_0) \rightarrow sigmoid \Rightarrow H_1 \rightarrow (V * H_1) \rightarrow relu \Rightarrow Y_1$
2. $(\mathrm{U} * X_2 + W * H_1) \rightarrow sigmoid \Rightarrow H_2 \rightarrow (V * H_2) \rightarrow relu \Rightarrow Y_2$
3. $(\mathrm{U} * X_3 + W * H_2) \rightarrow sigmoid \Rightarrow H_3 \rightarrow (V * H_3) \rightarrow relu \Rightarrow Y_3$

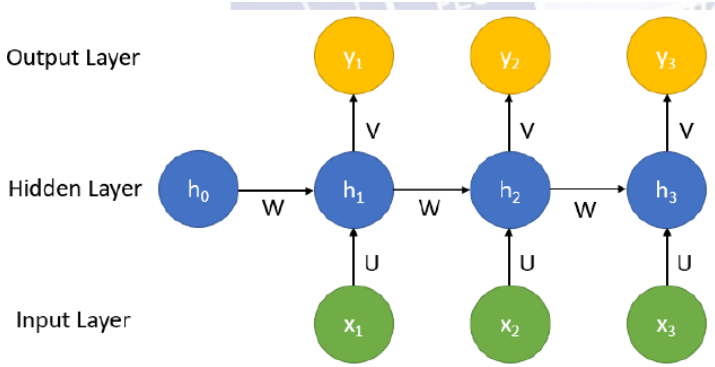


Fig 1. simple RNN

$$h_1 = f(U \cdot x_1 + W \cdot h_0)$$
$$y_1 = g(V \cdot h_1)$$
$$h_t = f(U \cdot x_t + W \cdot h_{t-1})$$
$$y_t = g(V \cdot h_t)$$

Fig 2. Formula

g(x)=max(0,x)

f(x)=σ (x)

Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

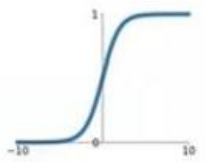


Fig 3.1. The sigmoid activation function

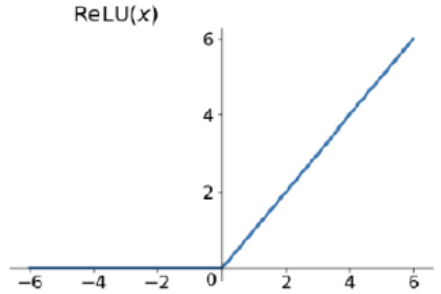


Fig 3.2. The ReLU activation function

Pipeline Design: Critical Issues

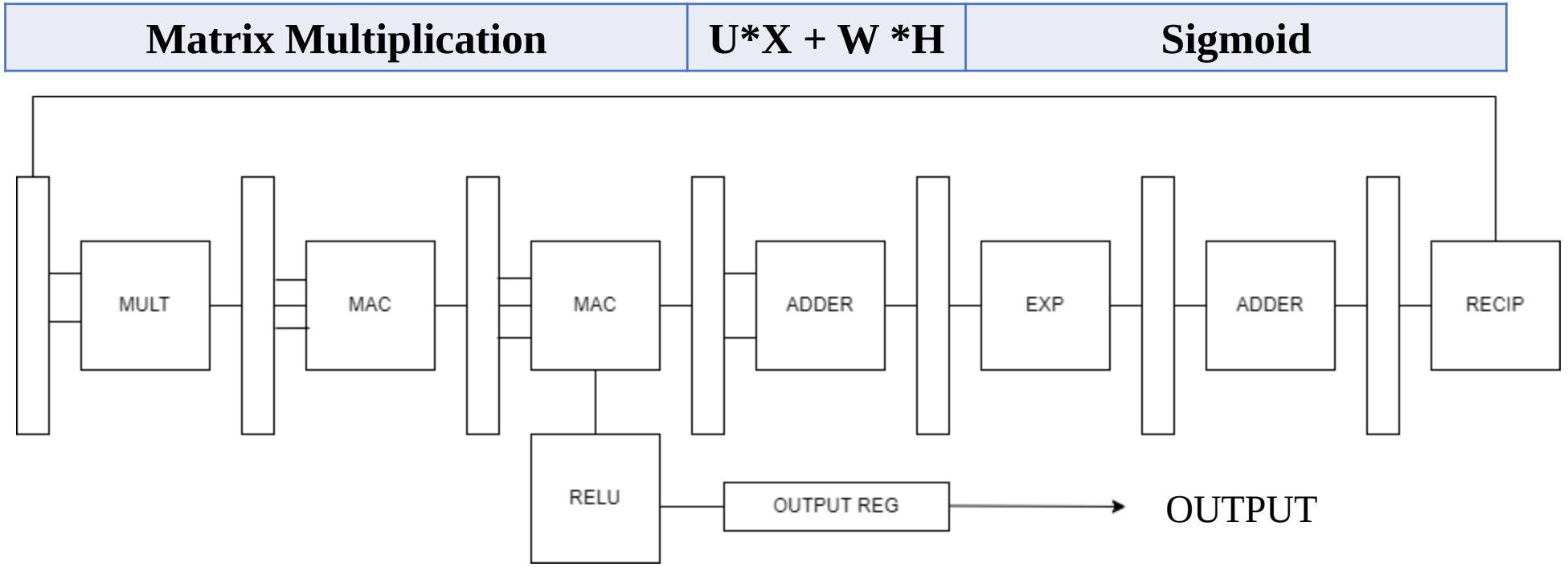
- Critical Issues: Timing and Area
  - Matrix Multiplication : Multiply 3 times, Add 2 times (DP3 vs MAC)

(IP)	DP3	MULT + MAC * 2
Cycle time	22ns	5ns~6ns
Cycle Count	1	3

  - Sigmoid: $Exp \rightarrow Add\ 1 \rightarrow Recip$ (1 Cycle vs 3 Cycles)

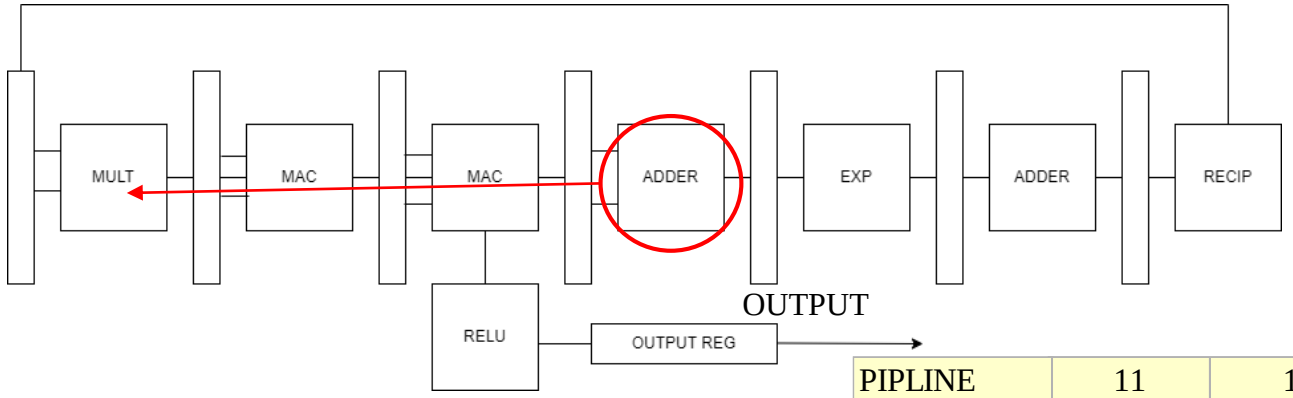
(IP)	EXP	ADDER	RECIP
Cycle time	16ns	5ns~6ns	15ns
Cycle Count	1	1	1

Pipeline Design: Stages



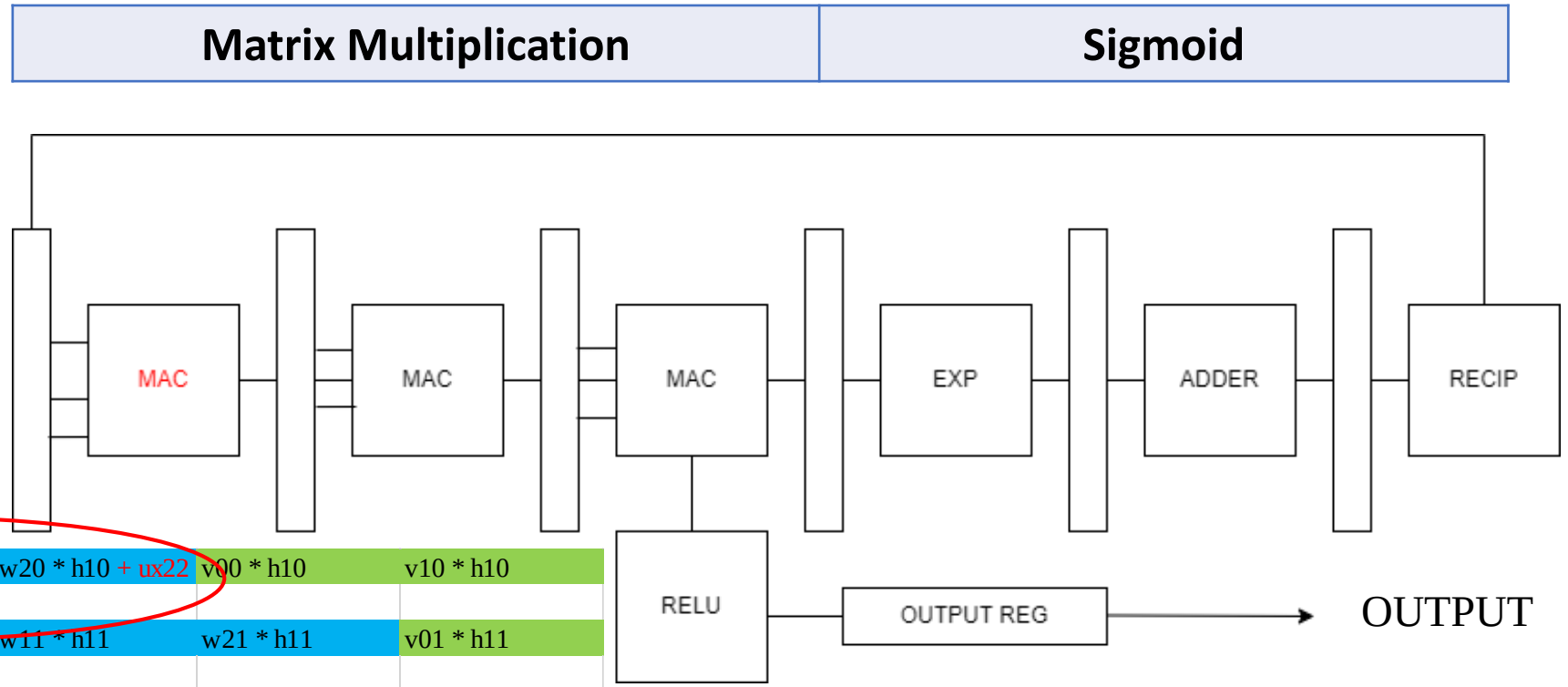
Cycle Count Reducing: Changing pipeline stages

Matrix Multiplication	U*X + W *H	Sigmoid



PIPLINE	11	12	13	14	15	16	17
Mult 1	w00 * h10	w10 * h10	w20 * h10	u20 * x30	v00 * h10	v10 * h10	v20 * h10
MAC1	u11 * x31	w01 * h11	w11 * h11	w21 * h11	u21 * x31	v01 * h11	v11 * h11
MAC2 + Relu	u02 * x32	u12 * x32	w02 * h12	w12 * h12	w22 * h12	u22 * x32	v02 * h12
	ux30	ux31	wh10	wh11	wh12	ux32	vh10 -> y10
Adder1				wh10 + ux20	wh11 + ux21	wh12 + ux22	
				wh10 + ux20	wh11 + ux21	wh12 + ux22	
Exp					wh10 + ux20	wh11 + ux21	wh12 + ux22
					ex20	ex21	ex22
Adder2	ex12 + 1					ex20 + 1	ex21 + 1
	ax12						ax20
Recip		ax12					a20
		rx12 -> h12					rx20 -> h20

Cycle Count Reducing: Changing pipeline stages



Mult adder 1	w00 * h10 + ux20	w10 * h10 + ux21	w20 * h10 + ux22	v00 * h10	v10 * h10
Mult adder 2	u21 * x31	w01 * h11	w11 * h11	w21 * h11	v01 * h11
Multa adder 3 + Relu	u12 * x32	u22 * x32	w02 * h12	w12 * h12	w22 * h12
	ux31	ux32	wh10 + ux20	wh11 + ux21	wh12 + ux22
Exp				wh10 + ux20	wh11 + ux21
				ex20	ex21
Add2	ex12 + 1				ex20 + 1
	ax12				ax20
Recip		ax12			
		rx12 -> h12			

Reduce Register: Reordering pipeline schedule

- Register Reducing
  - Ex. The result of (U $*$ X_2) and H_2 can store in X_2_reg. X_3_reg can store the result of (U $*$ X_3) and H_3 as well.

4	5	6	7	8	9
u10 * x10	u00 * x20	u10 * x20	u20 * x20	u20 * x10	u00 * x30
u01 * x11	u11 * x11	u01 * x21	u11 * x21	u21 * x21	u21 * x11
	u02 * x12	u12 * x12	u02 * x22	u12 * x22	u22 * x22
	ux10	ux11	ux20	ux21	ux22
		ux10	ux11		
		ex10	ex11		
			ex10 + 1	ex11 + 1	
			ax10	ax11	
				ax10	ax11
				rx10 -> h10	rx11 -> h11

Pipeline

PIPLINE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Mult adder 1			u00 * x10	u10 * x10	u00 * x20	u10 * x20	u20 * x20	u20 * x10	u00 * x30	u10 * x30	u20 * x30	w00 * h10 + ux20	w10 * h10 + ux21	w20 * h10 + ux22	v00 * h10
Mult adder 2				u01 * x11	u11 * x11	u01 * x21	u11 * x21	u21 * x21	u21 * x11	u01 * x31	u11 * x31	u21 * x31	w01 * h11	w11 * h11	w21 * h11
Multa adder 3 + Relu					u02 * x12	u12 * x12	u02 * x22	u12 * x22	u22 * x22	u22 * x12	u02 * x32	u12 * x32	u22 * x32	w02 * h12	w12 * h12
					ux10	ux11	ux20	ux21	ux22	ux12	ux30	ux31	ux32	wh10 + ux20	wh11 + ux21
Exp						ux10	ux11				ux12				wh10 + ux20
						ex10	ex11				ex12				ex20
Add2							ex10 + 1	ex11 + 1				ex12 + 1			
							ax10	ax11				ax12			
Recip								ax10	ax11				ax12		
								rx10 -> h10	rx11 -> h11				rx12 -> h12		

PIPLINE	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Mult adder 1	v10 * h10	v20 * h10	w00 * h20 + ux30	w10 * h20 + ux31	w20 * h20 + ux32	v00 * h20	v10 * h20	v20 * h20	v00 * h30	v10 * h30	v20 * h30			
Mult adder 2	v01 * h11	v11 * h11	v21 * h11	w01 * h21	w11 * h21	w21 * h21	v01 * h21	v11 * h21	v21 * h21	v01 * h31	v11 * h31	v21 * h31		
Multa adder 3 + Relu	w22 * h12	v02 * h12	v12 * h12	v22 * h12	w02 * h22	w12 * h22	w22 * h22	v02 * h22	v12 * h22	v22 * h22	v02 * h32	v12 * h32	v22 * h32	
	wh12 + ux22	vh10 -> y10	vh11 -> y11	vh12- y12	wh20 + ux30	wh21 + ux31	wh22 + ux32	vh20 -> y20	vh21 - > y21	vh22 -> y22	vh30 -> y30	vh31 -> y31	vh32 -> y32	
Exp	wh11 + ux21	wh12 + ux22				wh20 + ux30	wh21 + ux31	wh22 + ux32						
	ex21	ex22				ex30	ex31	ex32						
Add2	ex20 + 1	ex21 + 1	ex22 + 1				ex30 + 1	ex31 + 1	ex32 + 1					
	ax20	ax21	ax22				ax30	ax31	ax32					
Recip		a20	a21	a22				a30	a31	a32				
		rx20 -> h20	rx21 -> h21	rx22 -> h22				rx30 -> h30	rx31 -> h31	rx32 -> h32				
					OUTPUT	y10	y11	y12	y20	y21	y23	y30	y31	y32

Thanks for your attention!