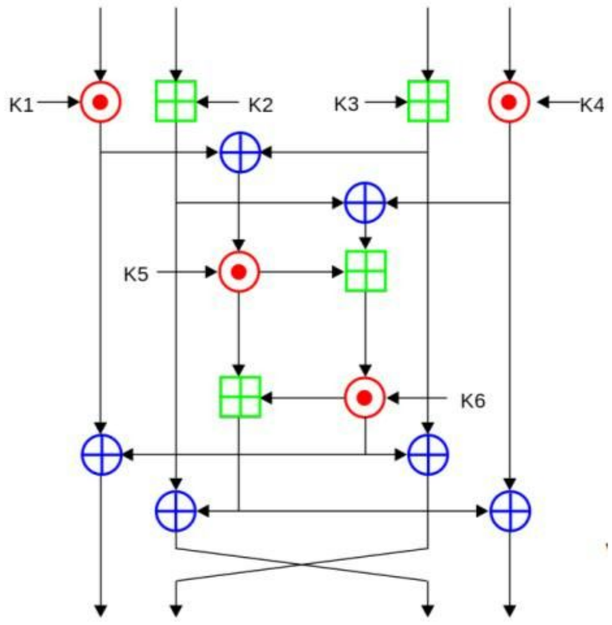


mult mod 2^16 + 1

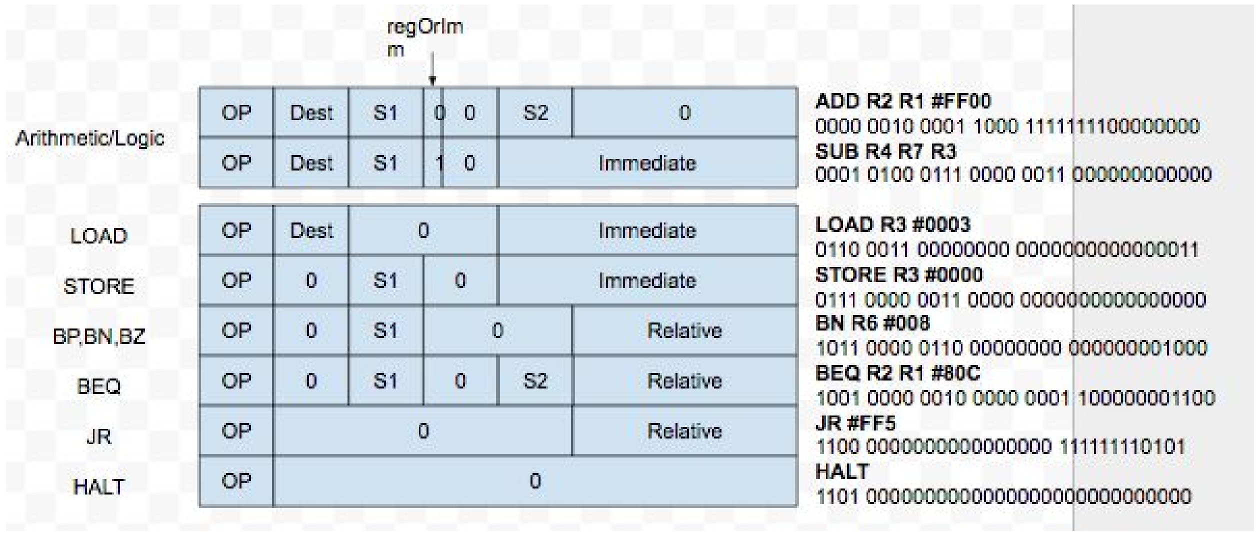
```
uint16 x
uint16 y
uint32 product
uint16 result

product = x*y
If LOW(product) >= HIGH(product)
        result = LOW(product) - HIGH(product)
else
        result = LOW(product) - HIGH(product) + 1
```

OP Codes

ADD =	0000
SUB =	0001
MUL =	0010
OR =	0011
AND =	0100
XOR =	0101
LOAD =	0110
STORE =	0111
BZ =	1000
BEQ =	1001
BP =	1010
BN =	1011
JR =	1100
HALT =	1101

Instruction encodings

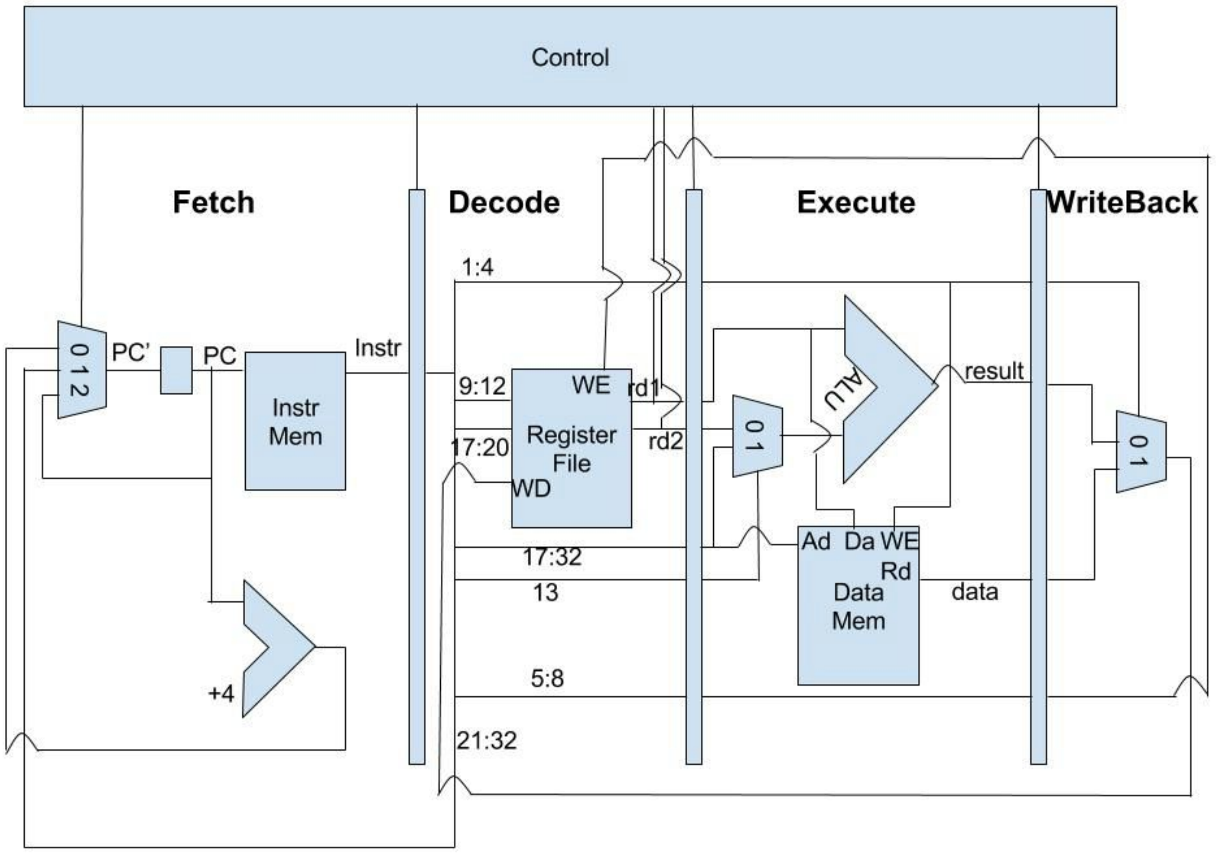


regOrImm

Arithmetic/Logic	OP	Dest	S1	0	0	S2	0
	OP	Dest	S1	1	0	Immediate	
LOAD	OP	Dest	0			Immediate	
STORE	OP	0	S1	0		Immediate	
BP,BN,BZ	OP	0	S1	0			Relative
BEQ	OP	0	S1	0		S2	Relative
JR	OP	0					Relative
HALT	OP	0					

ADD R2 R1 #FF00
0000 0010 0001 1000 1111111100000000
SUB R4 R7 R3
0001 0100 0111 0000 0011 000000000000
LOAD R3 #0003
0110 0011 00000000 0000000000000011
STORE R3 #0000
0111 0000 0011 0000 0000000000000000
BN R6 #008
1011 0000 0110 00000000 000000001000
BEQ R2 R1 #80C
1001 0000 0010 0000 0001 100000001100
JR #FF5
1100 0000000000000000 111111110101
HALT
1101 0000000000000000000000000000

Pipeline



Hazard Examples

F = fetch							
D = decode							
E = execute							
WB = write back							
RAW Hazard (R3)							
ADD R3 R2 R1	F	D	E	WB			
SUB R9 R4 R3		F	F	D	E	WB	
OR R12 R10 R11				F	D	E	WB
Branch not taken							
BEQ R4 R5 #00C	F	D	E	WB			
ADD R1 R2 R3		F	D	E	WB		
Branch taken							
BEQ R1 R2 #00C	F	D	E	WB			
ADD R5 R3 R4		F	-	-	-		
XOR R7 R6 #00FF							
AND R5 R3 R4			F	D	E	WB	
Multiplication							
MUL R3 R1 R2	F	D	E	WB (R3)	WB (R4)		
ADD R7 R5 R6		F	F	D	E	WB	
Multiplication with RAW Hazard							
MUL R3 R1 R2	F	D	E	WB (R3)	WB (R4)		
ADD R8 R9 R3		F	F	D	E	WB	

				Write-back first half cycle, so no extra stall needed			
Multiplication with RAW Hazard for second 16 bits							
MUL R3 R1 R2	F	D	E	WB (R3)	WB (R4)		
ADD R8 R9 R4		F	F	F	D	E	WB

Comparison with other implementations

	Transistor Count	**Throughput**	**System Clock Speed**	**Year**	**Throughput/Transistor Count**
Bibliography #3	190K	64Mbps	8MHz	2001	336
Bibliography #4	251K	177Mbps	25MHz	1993	705
Me	1.3M	182Mbps	1GHz	2016	140

Bibliography

1. Leong, M. P., Cheung, O. Y., Tsoi, K. H., & Leong, P. H. W. (2000). A bit-serial implementation of the international data encryption algorithm IDEA. In*Field-Programmable Custom Computing Machines, 2000 IEEE Symposium on* (pp. 122-131). IEEE.
2. Kim, S., & Cho, K. (2010). Design of high-speed modified booth multipliers operating at GHz ranges. *World Academy of Science, Engineering and Technology*, *61*, 1-4.
3. Sklavos, N., & Koufopavlou, O. (2001). Asynchronous low power vlsi implementation of the international data encryption algorithm. In *Electronics, Circuits and Systems, 2001. ICECS 2001. The 8th IEEE International Conference on* (Vol. 3, pp. 1425-1428). IEEE.
4. Zimmermann, R., Curiger, A., Bonnenberg, H., Kaeslin, H., Felber, N., & Fichtner, W. (1994). A 177 Mb/s VLSI implementation of the international data encryption algorithm. *Solid-State Circuits, IEEE Journal of*, *29*(3), 303-307.
5. Steinhaus, M., Kolla, R., Larriba-Pey, J. L., Ungerer, T., & Valero, M. (2001, June). Transistor count and chip-space estimation of simplescalar-based microprocessor models. In *Proceedings of the Workshop on Complexity-Effective Design* (pp. 1-15).

6. Hennessy, J. & Patterson, D. (2006). Computer Architecture: A Quantitative Approach (5th Edition) Prentice-Hall.