

# The Distributed Approach

Daniel Roop

Tyler Hunt

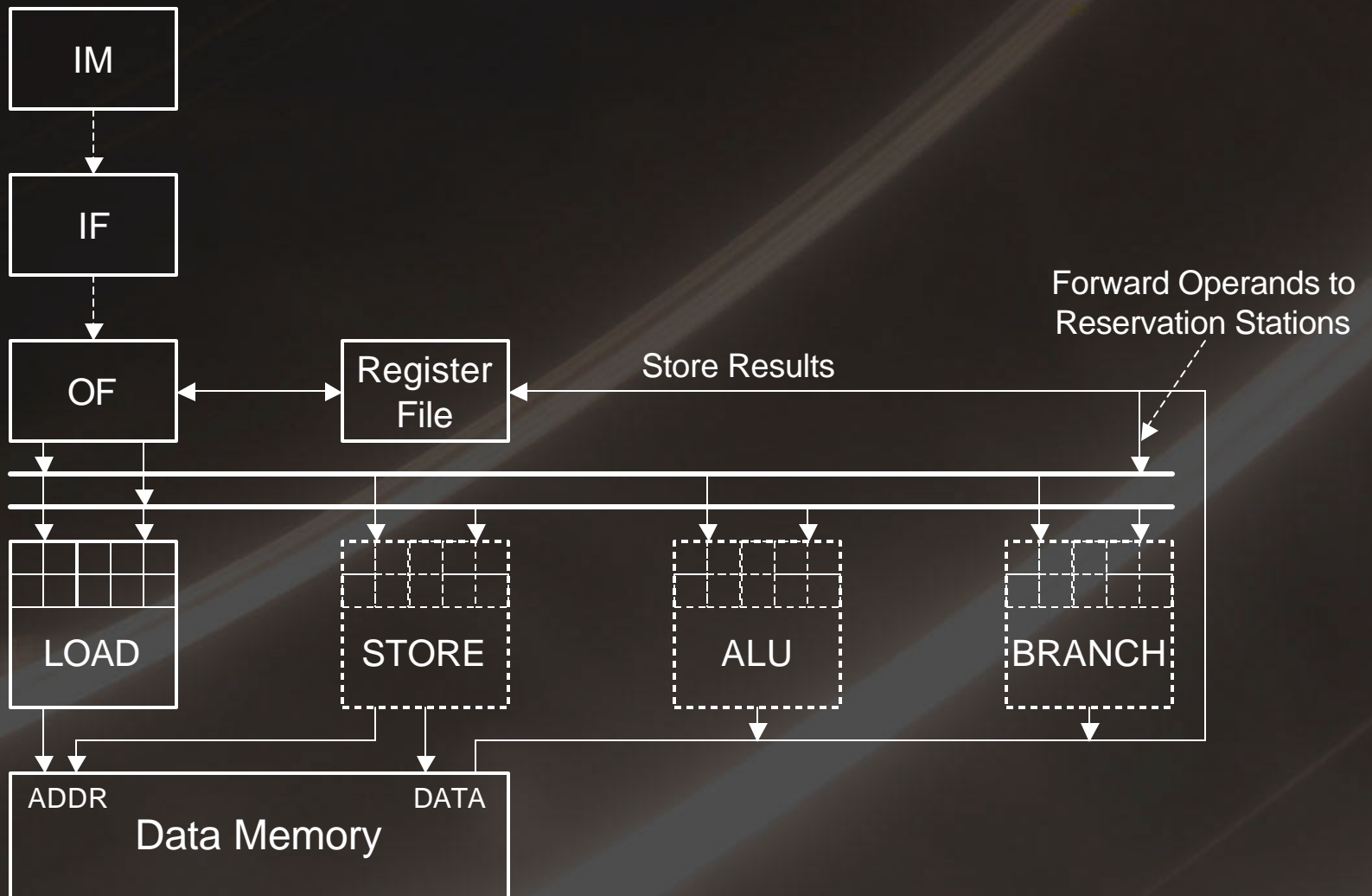
# Centralized Instruction Windows

- It holds all pending instructions
- When all the operands arrive to the instruction the instruction is routed to the appropriate functional unit for execution
- More than bus is provided to allow the issue of multiple instructions
- When a register (in register file) is updated, the instruction window is accessed and any matching register (in the instruction) is replaced by the value of the register

# Distributed Instruction Windows

- The instruction buffers are called “reservation stations”
- The reservation stations are placed at the front of each functional unit

# Distributed Instruction Windows



# Register Renaming

- Example:

MULT R <sub>1</sub> , R <sub>2</sub> , R <sub>4</sub>	R <sub>1</sub> ← R <sub>2</sub> × R <sub>4</sub>
ADD R <sub>2</sub> , R <sub>6</sub> , 1	R <sub>2</sub> ← R <sub>6</sub> + 1
ADD R <sub>1</sub> , R <sub>4</sub> , R <sub>5</sub>	R <sub>1</sub> ← R <sub>4</sub> + R <sub>5</sub>

- By introducing register renaming: R<sub>1</sub><sup>\*</sup> and R<sub>2</sub><sup>\*</sup> will be used to avoid data dependencies

MULT R <sub>1</sub> , R <sub>2</sub> , R <sub>4</sub>
ADD R <sub>2</sub> <sup>*</sup> , R <sub>6</sub> , 1
ADD R <sub>1</sub> <sup>*</sup> , R <sub>4</sub> , R <sub>5</sub>

# Register Renaming Example

ADD R<sub>3</sub>, R<sub>4</sub>, R<sub>5</sub>

LOAD R<sub>7</sub>, (R<sub>3</sub>)

SUB R<sub>3</sub>, R<sub>12</sub>, R<sub>11</sub>

STORE (R<sub>15</sub>), R<sub>3</sub>

ADD R<sub>3</sub>, R<sub>4</sub>, R<sub>5</sub>

LOAD R<sub>7</sub>, (R<sub>3</sub>)

SUB R<sub>HW20</sub>, R<sub>12</sub>, R<sub>11</sub>

STORE (R<sub>15</sub>), R<sub>HW20</sub>

## Before Register Renaming

ADD	IF	OF	EX	MEM	WB						
LOAD		IF	<del>STALL</del>	<del>STALL</del>	<del>STALL</del>	OF	EX	MEM	WB		
SUB			IF	OF	EX	MEM	WB				
STORE				IF	<del>STALL</del>	<del>STALL</del>	<del>STALL</del>	OF	EX	MEM	WB

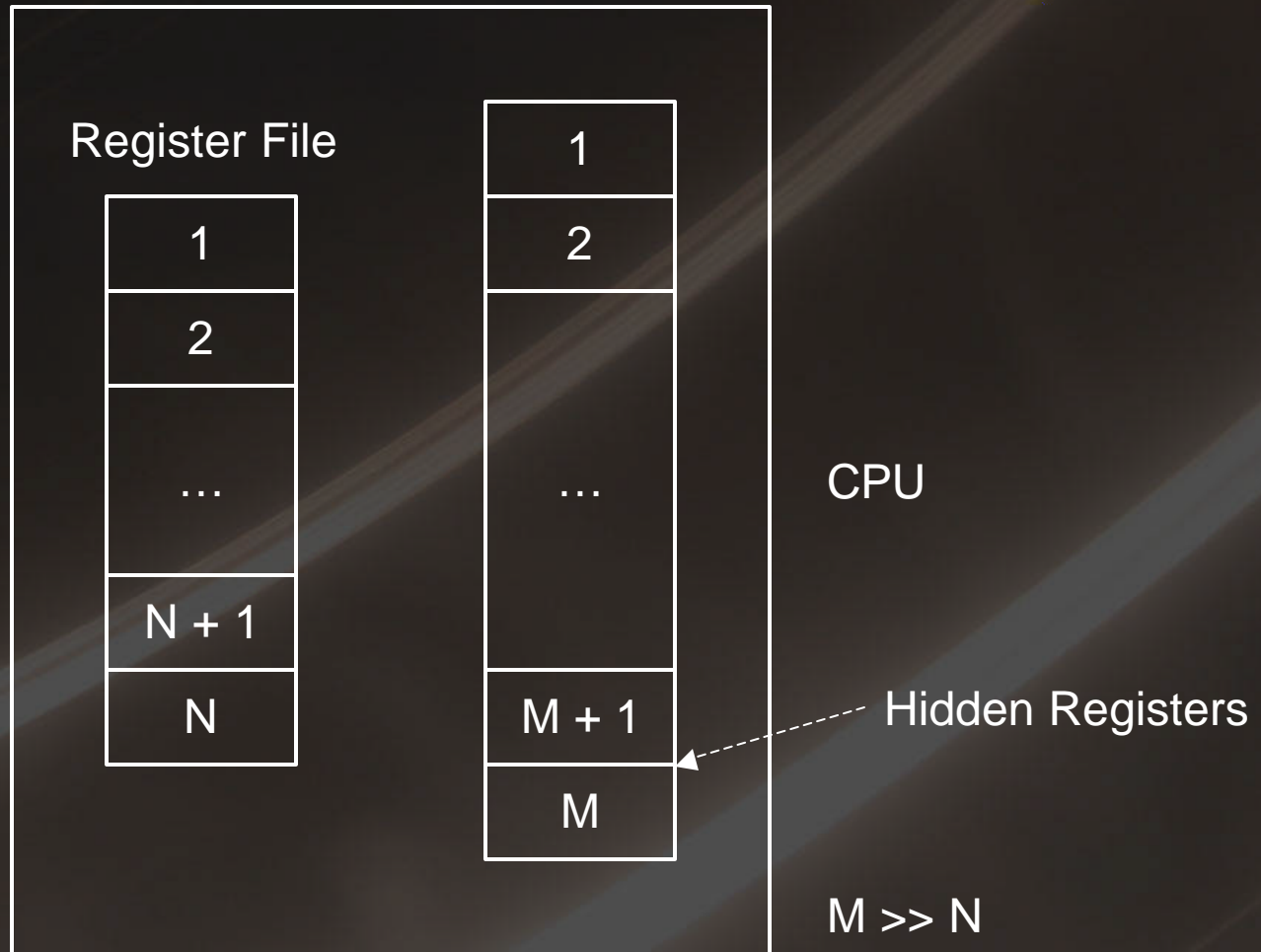
## After Register Renaming

ADD	IF	OF	EX	MEM	WB						
LOAD		IF	<del>STALL</del>	<del>STALL</del>	<del>STALL</del>	OF	EX	MEM	WB		
SUB	IF	OF	EX	MEM	WB						
STORE		IF	<del>STALL</del>	<del>STALL</del>	<del>STALL</del>	OF	EX	MEM	WB		

# Register Renaming Summary

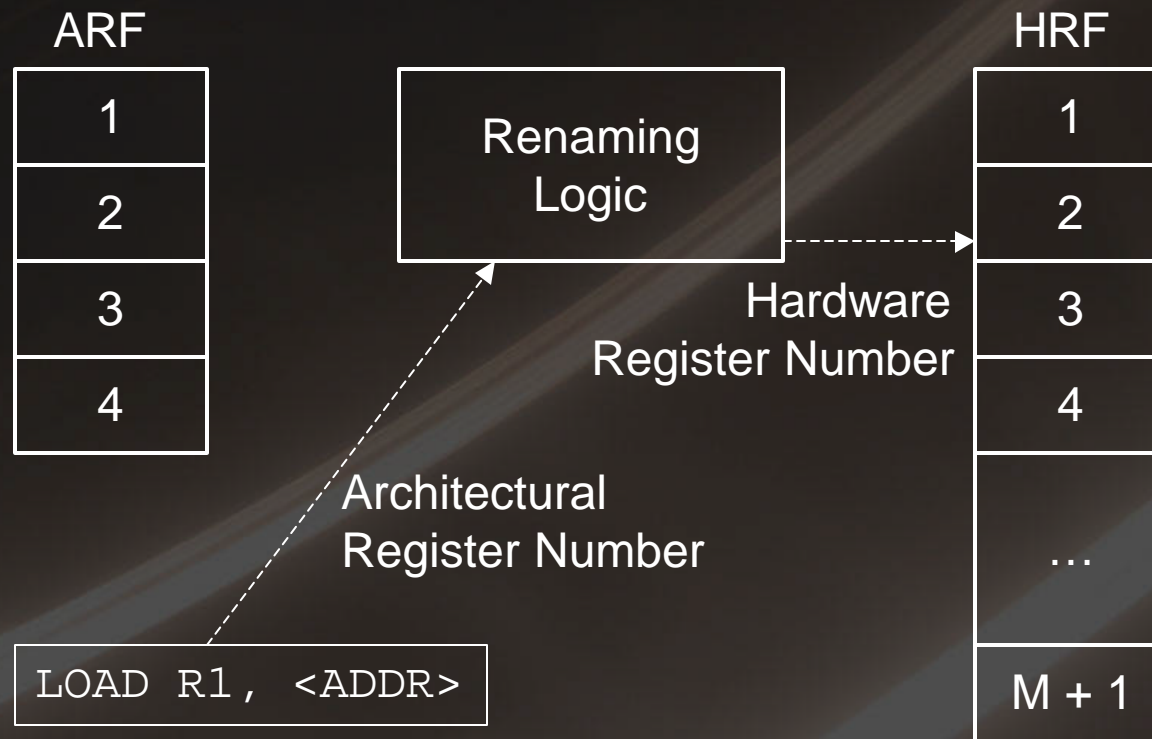
- We have a set of registers in the processor
- The number of registers is limited
- This technique dynamically assigns values to new registers

# Hidden Registers





# Architectural Register File



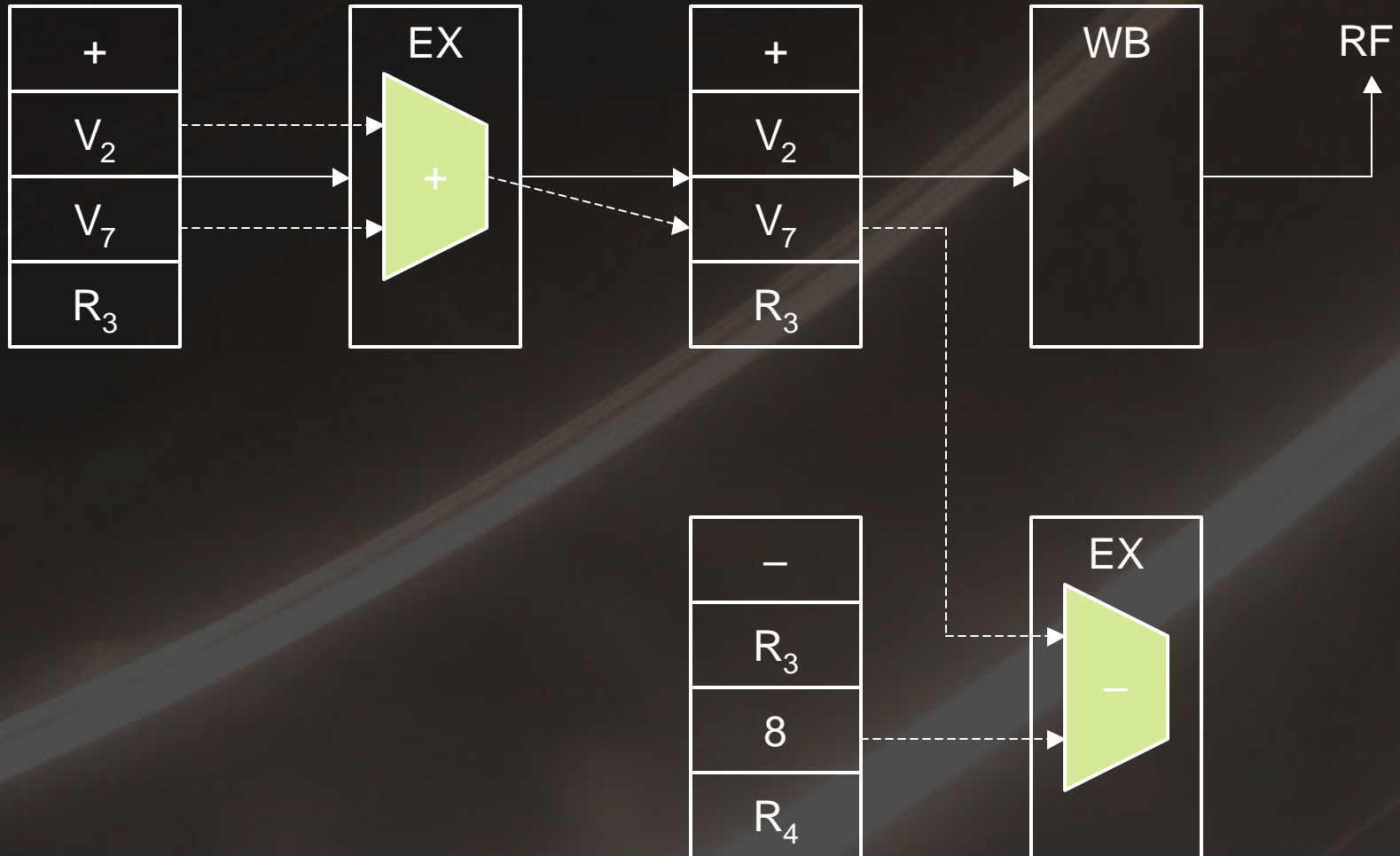
# Forwarding

- Forwarding is a technique that allows us to pass results of one instruction directly to another instruction
- Example:

ADD $R_3, R_2, R_7$	$R_3 \leftarrow R_2 + R_7$	IF	OF	EX	WB	
SUB $R_4, R_3, 8$	$R_4 \leftarrow R_3 - 8$		IF	OF	EX	WB



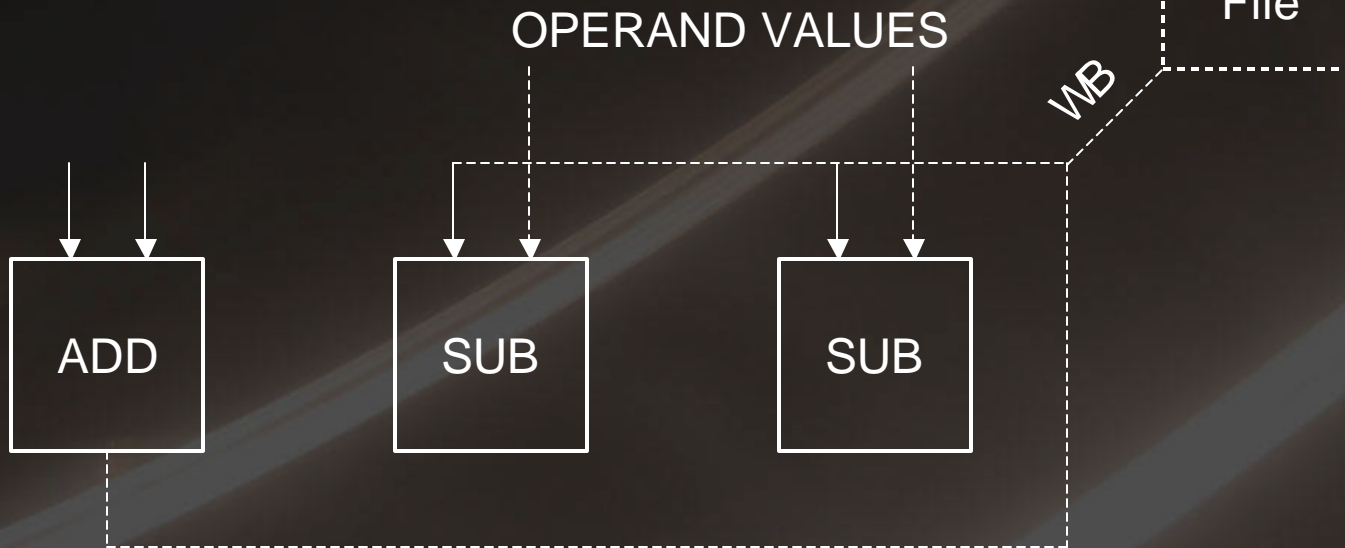
# Forwarding Example (Cont.)



# Forwarding

(Using Multiple Functional Units)

```
ADD R3, R2, R7  
SUB R4, R3, 8  
SUB R5, R3, 4
```



This model was used in the IBM 360 Model 91 as reported by Tomasulo (1969)