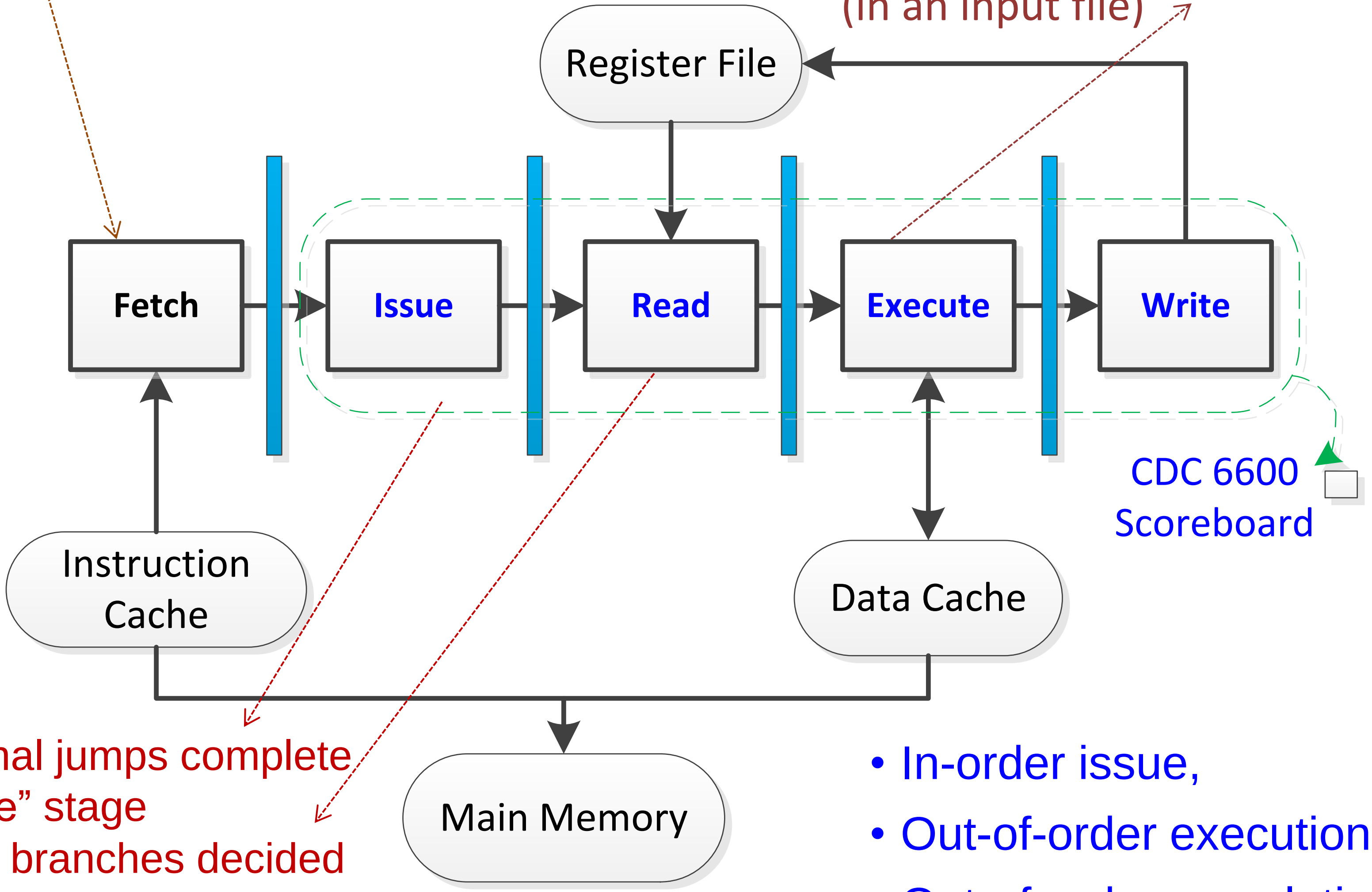
When a fetched instruction cannot be issued, the next instruction cannot be fetched.

Multiple functional units whose numbers are provided as configuration parameters (in an input file)



- Unconditional jumps complete in the “Issue” stage
- Conditional branches decided in the “Operand Read” stage.
- “not-taken prediction” will be used in “Fetch” stage.

- In-order issue,
- Out-of-order execution
- Out-of-order completion
- No data bypassing

Instruction Class	Instruction Mnemonic
Data Transfers	LW, SW, L.D, S.D
Arithmetic/ logical	DADD, DADDI, DSUB, DSUBI, AND, ANDI, OR, ORI, ADD.D, MUL.D, DIV.D, SUB.D
Control	J, BEQ, BNE
Special purpose	HLT (to stop fetching new instructions)

The table below shows the number of cycles each instruction takes in the EX stage.

Instructions	Number of Cycles in "Execute" Stage
HLT, J	0 Cycles (finish in issue stage)
BEQ, BNE	0 Cycle (finish in Read stage)
DADD, DADDI, DSUB, DSUBI, AND, ANDI, OR, ORI	1 Cycle
LW, SW, L.D, S.D	1 Cycle + memory access time (D-Cache)
ADD.D, SUB.D	2 Cycles
MUL.D	30 Cycles
DIV.D	50 Cycles

Example

inst.txt

```
GG:   L.D  F1,  4(R4)
      L.D  F2,  8(R5)
      ADD.D   F4,   F6,  F2
      SUB.D    F5,   F7,  F1
      MUL.D   F6,   F1, F5
      ADD.D   F7,  F2,  F6
      ADD.D   F6,  F1,  F7
      DADDI   R4,   R4,  2
      DADDI   R5,   R5,  2
      DSUB    R1,   R1,  R2
      BNE      R1,   R3,  GG
      HLT
```

config.txt

```
2
2
1
```

FT Adders

FT Multipliers

FT Division units

Example: Without Memory Hierarchy

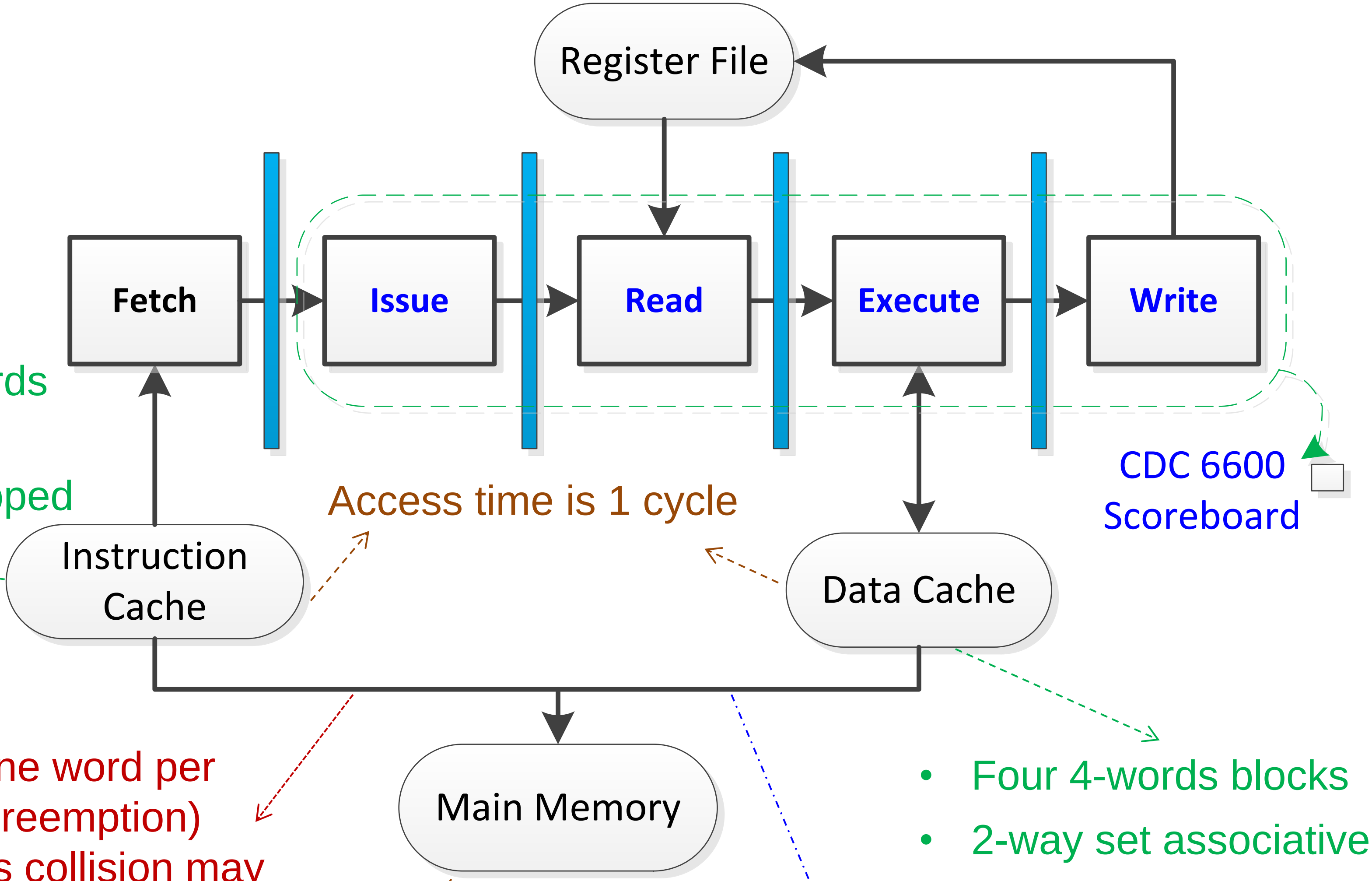
(1st iteration)

Instruction	Fetch	Issue	Read	Exec	Write	RAW	WAR	WAW	Struct
GG: L.D F1, 4(R4)	1	2	3	6	7	N	N	N	N
L.D F2, 8(R5)	2	8	9	12	13	N	N	N	Y
ADD.D F4, F6, F2	8	9	**14**	16	17	**Y**	N	N	N
SUB.D F5, F7, F1	9	10	11	13	14	N	N	N	N
MUL.D F6, F1, F5	10	11	**15**	45	46	**Y**	N	N	N
ADD.D F7, F2, F6	11	15	**47**	49	50	**Y**	N	N	Y
ADD.D F6, F1, F7	15	47	**51**	53	54	**Y**	N	Y	N
DADDI R4, R4, 2	47	48	49	50	51	N	N	N	N
DADDI R5, R5, 2	48	52	53	54	55	N	N	N	Y
DSUB R1, R1, R2	52	56	57	58	59	N	N	N	Y
BNE R1, R3, GG	56	57	**60**			**Y**	N	N	N
HLT	57	58				N	N	N	N

Example: Without Memory Hierarchy

(2nd iteration)

Instruction	Fetch	Issue	Read	Exec	Write	RAW	WAR	WAW	Struct
GG: L.D F1, 4(R4)	61	62	63	66	67	N	N	N	N
L.D F2, 8(R5)	62	68	69	72	73	N	N	N	Y
ADD.D F4, F6, F2	68	69	74	76	77	Y	N	N	N
SUB.D F5, F7, F1	69	70	71	73	74	N	N	N	N
MUL.D F6, F1, F5	70	71	75	105	106	Y	N	N	N
ADD.D F7, F2, F6	71	75	107	109	110	Y	N	N	Y
ADD.D F6, F1, F7	75	107	111	113	114	Y	N	Y	Y
DADDI R4, R4, 2	107	108	109	110	111	N	N	N	N
DADDI R5, R5, 2	108	112	113	114	115	N	N	N	Y
DSUB R1, R1, R2	112	116	117	118	119	N	N	N	Y
BNE R1, R3, GG	116	117	120			Y	N	N	N
HLT	117	118				N	N	N	N



CDC 6600 Scoreboard

- Four 4-words blocks
- Direct mapped

Access time is 1 cycle

- Four 4-words blocks
- 2-way set associative
- Write-back strategy
- Write-allocate policy

- Fetching one word per cycle (no preemption)
- Bus access collision may take place ➔ blocking
- Simultaneous access requests are resolved in favor of the I-cache

Access time is 3 cycles

In case there are an I-Cache miss and a D-Cache miss happens in the same cycle, priority will be given to I-Cache

Example: With Memory Hierarchy

(1st iteration)

Instruction	Fetch	Issue	Read	Exec	Write	RAW	WAR	WAW	Struct
GG: L.D F1, 4(R4)	12	13	14	28	29	N	N	N	N
L.D F2, 8(R5)	13	30	31	56	57	N	N	N	Y
ADD.D F4, F6, F2	30	31	**58**	60	61	**Y**	N	N	N
SUB.D F5, F7, F1	31	32	33	35	36	N	N	N	N
MUL.D F6, F1, F5	43	44	45	75	76	N	N	N	N
ADD.D F7, F2, F6	44	45	**77**	79	80	**Y**	N	N	N
ADD.D F6, F1, F7	45	77	**81**	83	84	**Y**	N	Y	Y
DADDI R4, R4, 2	77	78	79	80	81	N	N	N	N
DADDI R5, R5, 2	89	90	91	92	93	N	N	N	N
DSUB R1, R1, R2	90	94	95	96	97	N	N	N	Y
BNE R1, R3, GG	94	95	**98**			**Y**	N	N	N
HLT	95	96							

Example: With Memory Hierarchy

(2nd iteration)

Instruction	Fetch	Issue	Read	Exec	Write	RAW	WAR	WAW	Struct
GG: LD F1, 4(R4)	99	100	101	115	116	N	N	N	N
LD F2, 8(R5)	100	117	118	132	133	N	N	N	Y
ADD.D F4, F6, F2	117	118	**134**	136	137	**Y**	N	N	N
SUB.D F5, F7, F1	118	119	120	122	123	N	N	N	N
MUL.D F6, F1, F5	119	120	**124**	154	155	**Y**	N	N	N
ADD.D F7, F2, F6	120	124	**156**	158	159	**Y**	N	N	Y
ADD.D F6, F1, F7	124	156	**160**	162	163	**Y**	N	Y	Y
DADDI R4, R4, 2	156	157	158	159	160	N	N	N	N
DADDI R5, R5, 2	157	161	162	163	164	N	N	N	Y
DSUB R1, R1, R2	161	165	166	167	168	N	N	N	Y
BNE R1, R3, GG	165	166	169			Y	N	N	N
HLT	166	167				N	N	N	N

Output

	clock cycle that instruction leaves each stage					Hazards that caused stalls			
Instruction	Fetch	Issue	Read	Exec	Write	RAW	WAR	WAW	Struct
GG: LD F1, 4(R4)	12	13	14	28	29	N	N	N	N
LD F2, 8(R5)	13	30	31	56	57	N	N	N	Y
:									
:									
DSUB R1, R1, R2	161	165	166	167	168	N	N	N	Y
BNE R1, R3, GG	165	166	169			Y	N	N	N
HLT	166	167				N	N	N	N

Cycle number of last stage (memory for load/store)

A branching instruction terminates in “Read” stage and does not have entries in the Exec and Write stages.

Total number of access requests for instruction cache: 24

Number of instruction cache hits: 21

Total number of access requests for data cache: 8

Number of data cache hits: 4