Section 4.1
Microsteps, hard wired

Implementing instructions using
Micro steps

Level 1

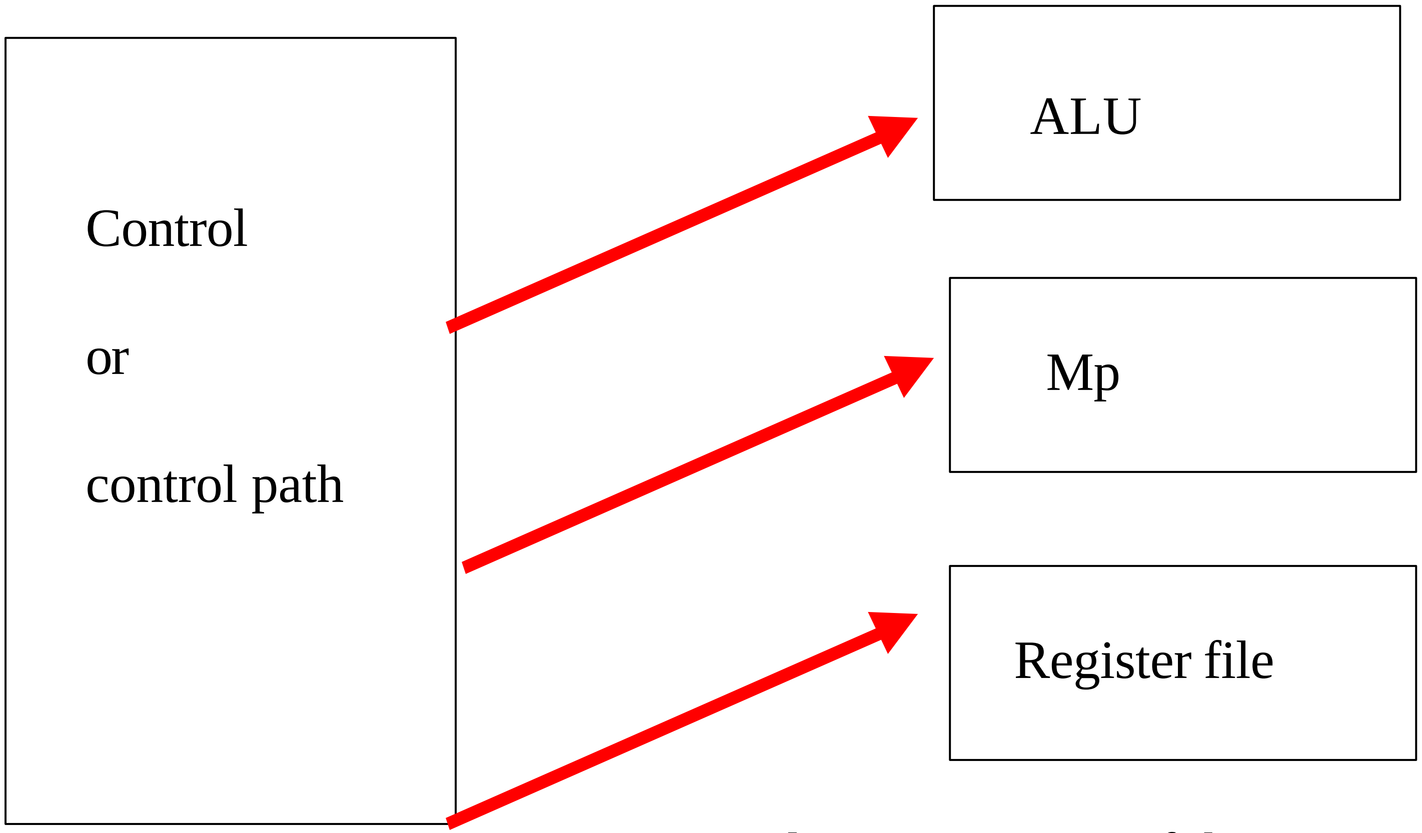
The purpose of the Control Path

Generate

the right sequence of control signals

to execute the instruction

The purpose of the Control



Hardware resources of the Data Path

Synchronous wired control

- Typical microsteps:
    - gate c(R1) to R2
    - start ALU
    - start Mp read cycle
    - seize bus

Control’s function, restated

- Generate time sequences of microsteps

- How?

computation

- the next micro step to be done depends on
  - which instruction we're executing (op code or order)   and
  - what time it is, i.e. which step we're doing (τ)

computation

- I.e.

- Next micro step = f(order, τ)

- control's purpose is to calculate this function, where

  τ ε 1,2,3, . . . , n

computation

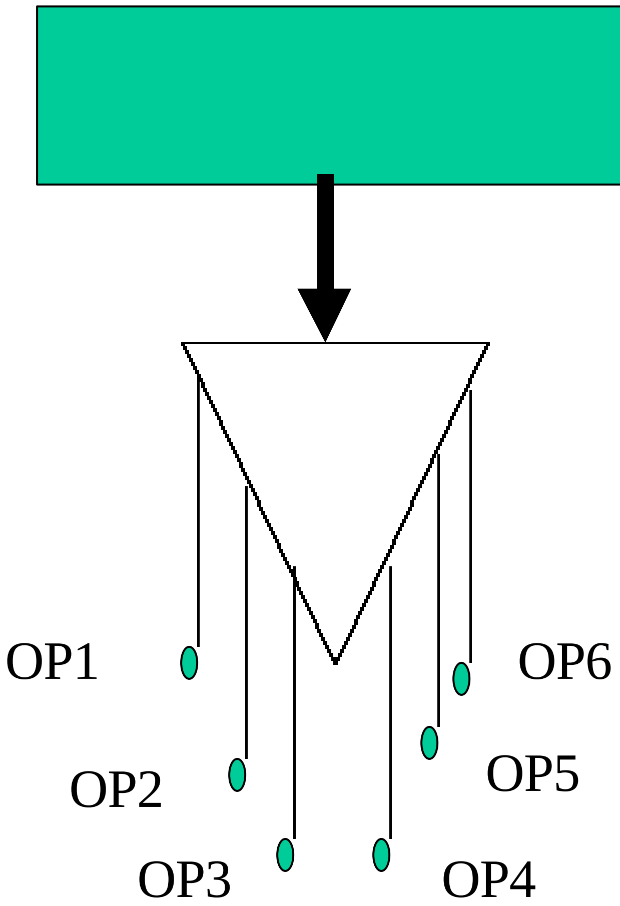
- Specifying function f tabular-ly
  - order1 --> μ step (1,1)
    - » μ step (1,2)
    - » μ step (1,3)
    - » .
    - » .
    - » μstep (1,n1)

computation

- Ditto for order2, . . . , order $_m$

- How to compute
  - “order” and τ in hardware?

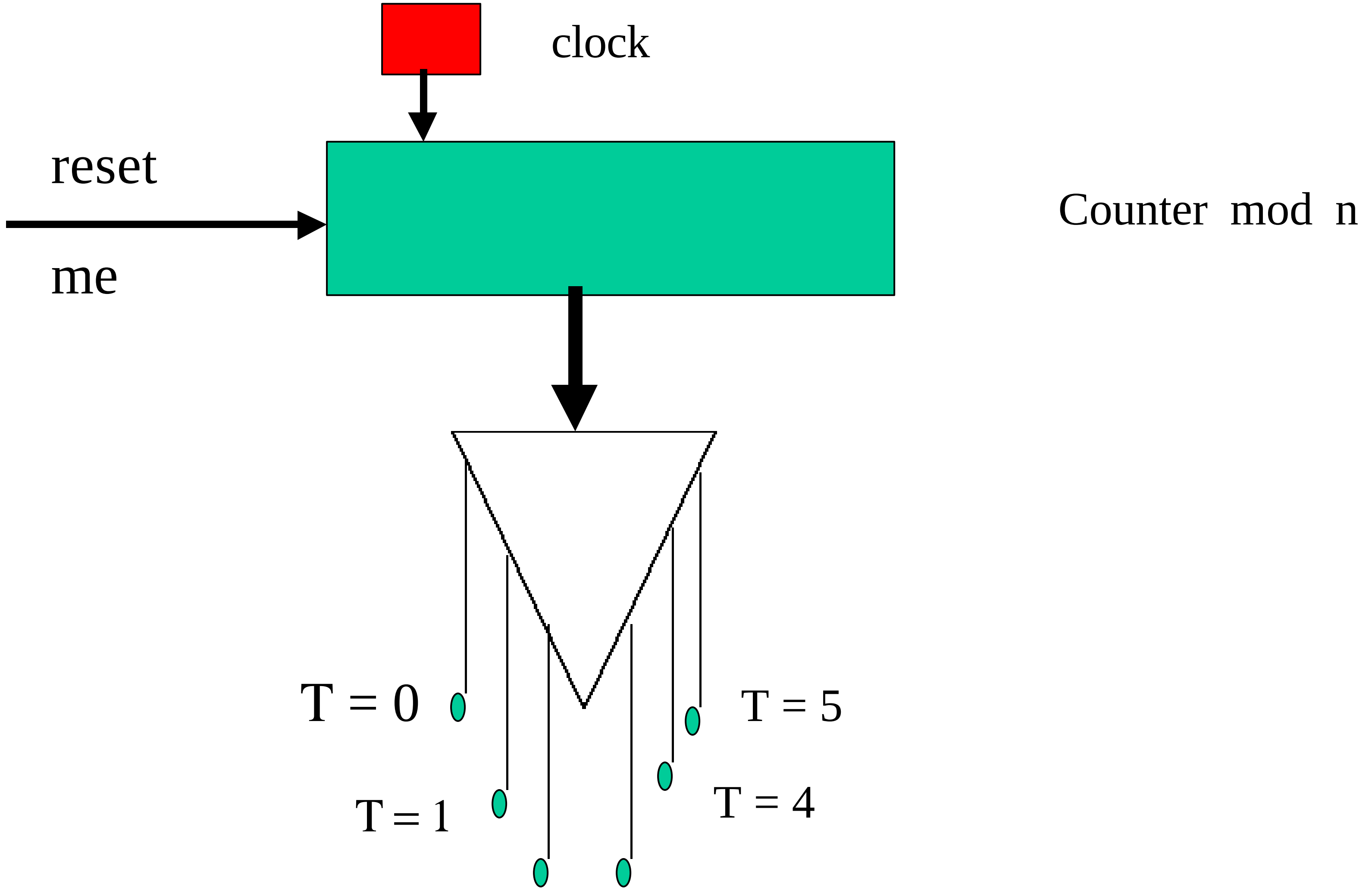
Computing “which order ?”



IOP, part of IR

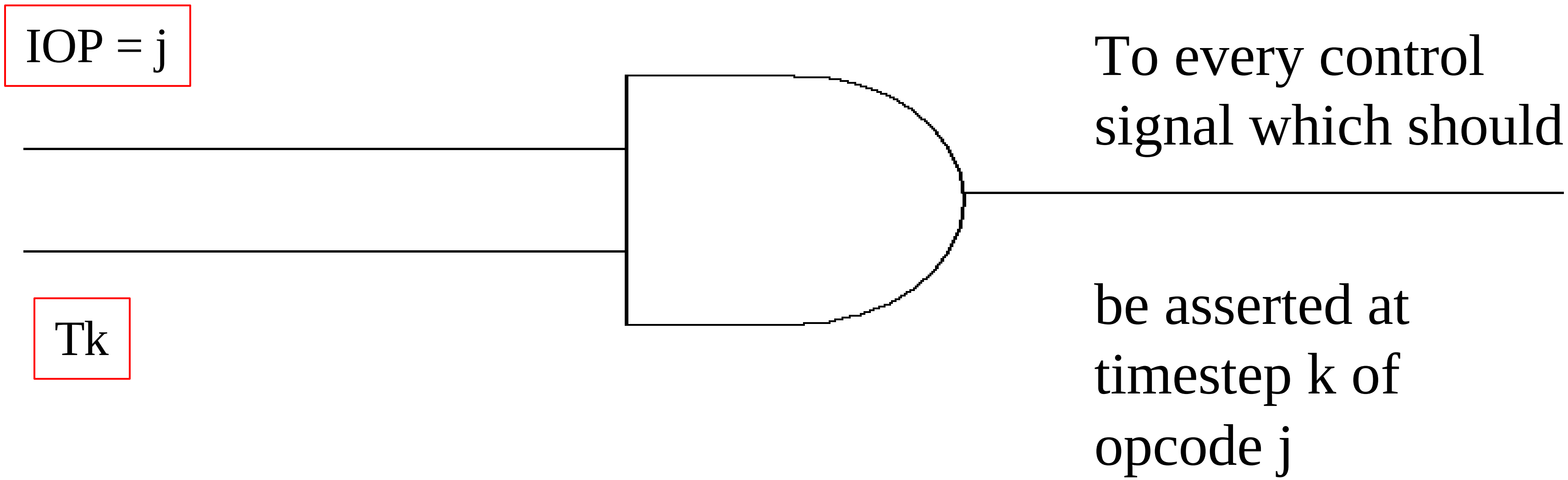
1/m decoder
$m = 2^k$ where
$k = \text{len(IOP)}$

Computing what time is it?
Value of τ



Putting it all together - a control

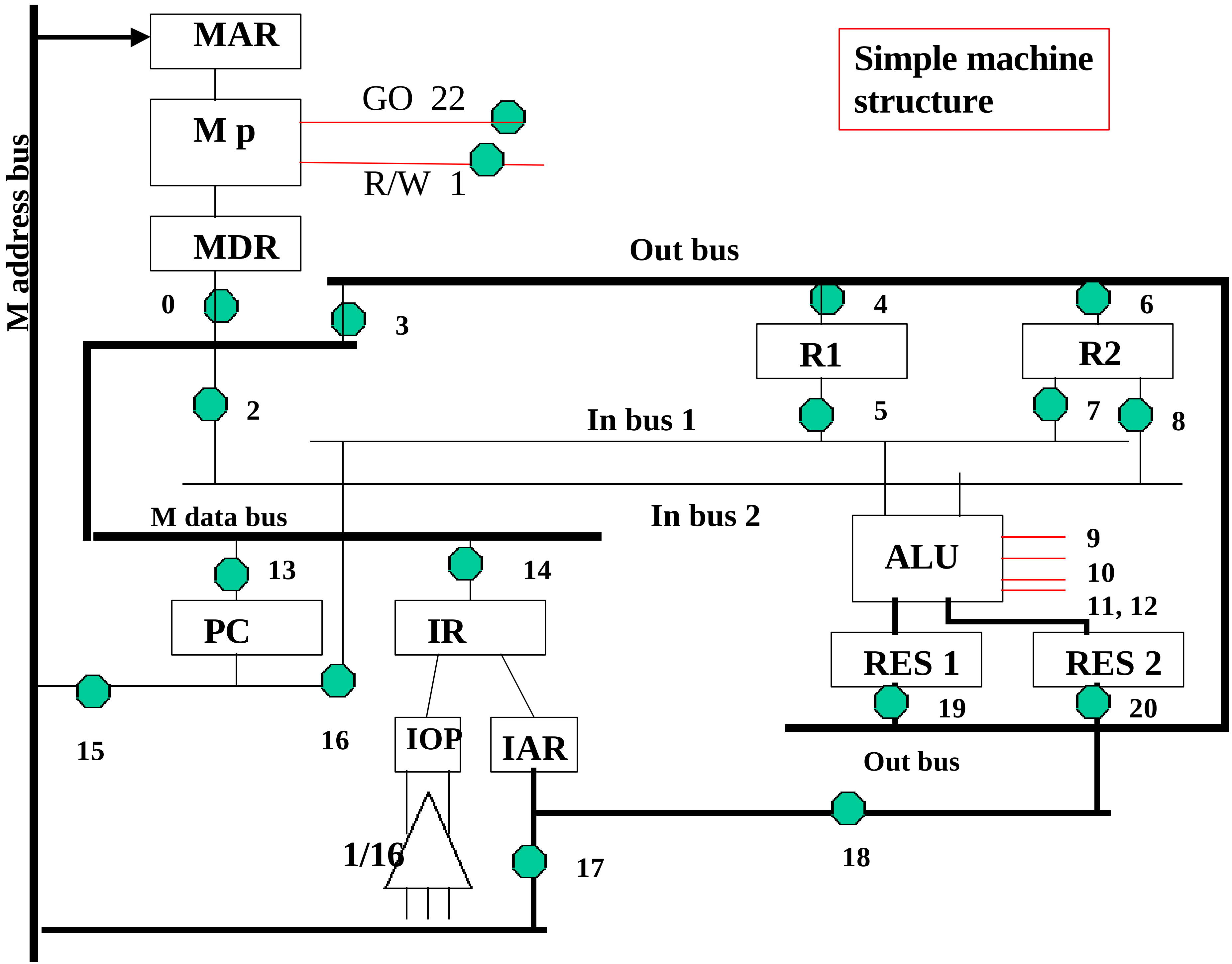
T1 means the same as tau = 1


IOP = j
Tk
To every control
signal which should
be asserted at
timestep k of
opcode j

Control in action: SimpleMachine

SimpleMachine has

- 2 general registers
- 16 opcodes
- fixed wordlength
- all instructions 1 word long
- same control problem as real machines



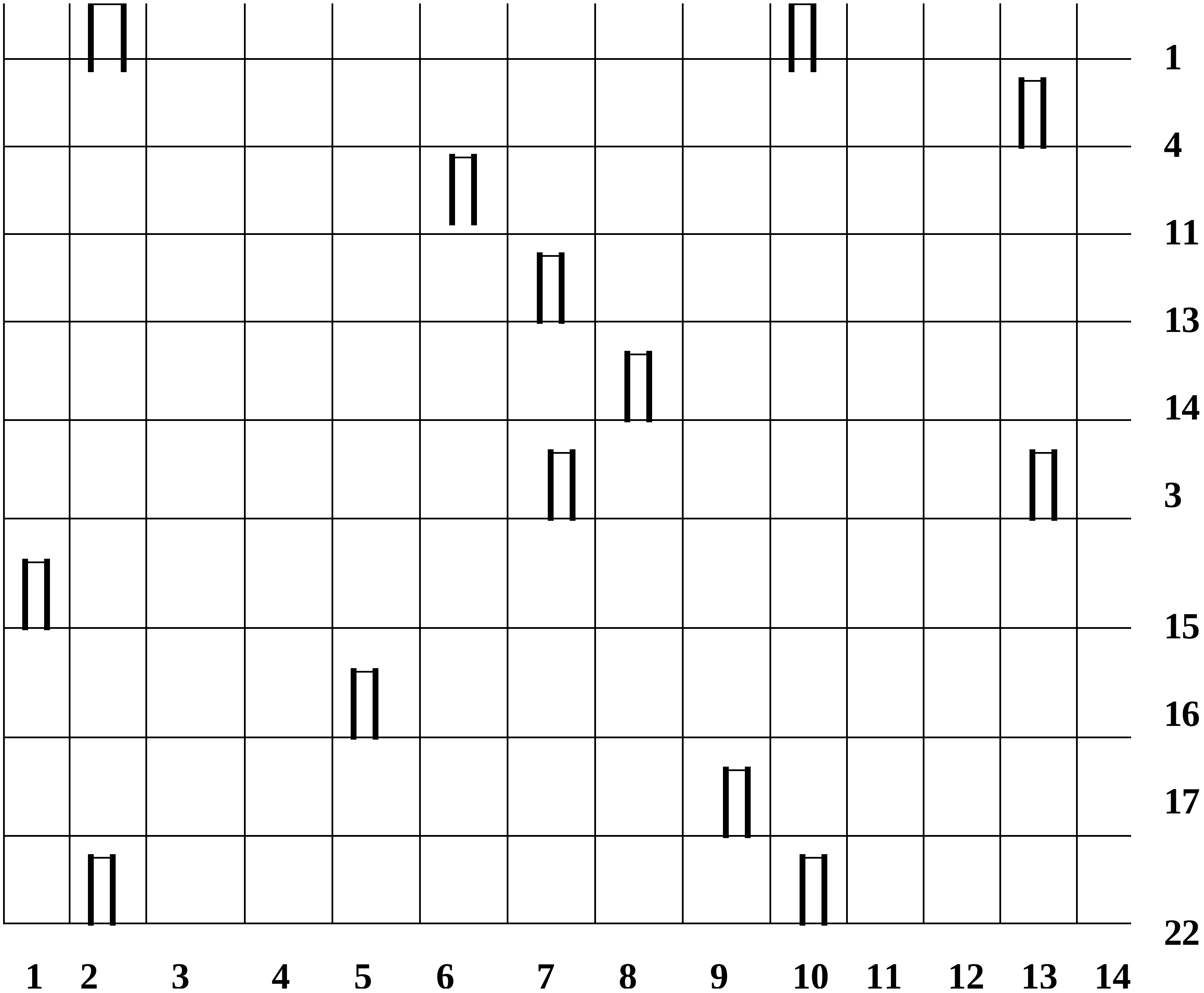
Simple machine structure
M address bus
MAR
M p
GO 22
R/W 1
MDR
Out bus
0
3
4
6
R1
R2
2
5
7
8
In bus 1
In bus 2
M data bus
ALU
9
10
11, 12
13
14
PC
IR
RES 1
RES 2
19
20
15
16
IOP
IAR
Out bus
1/16
17
18

Example: LR1 X

R1 <-- c(Address X)

Microsteps:

micro steps	**gates closed**	**t**
MAR <-- c(PC)	15	1
MEMREAD	1, 22	**2**
inbus1 <-- c(PC)	**16**	**5**
increment	**11**	**6**
PC <-- c(RES1)	**19,3,13**	**7**
IR <-- c(MBR)	**0, 14**	**8**
/* end of fetch phase */		
MAR <-- c(IAR)	**17**	**9**
MEMREAD		
R1 <-- c(MBR)	**0,3,4**	**13**


1
4
11
13
14
3
15
16
17
22
1 2 3 4 5 6 7 8 9 10 11 12 13 14

Conventional Control - Wired logic
Fill in the Blanks