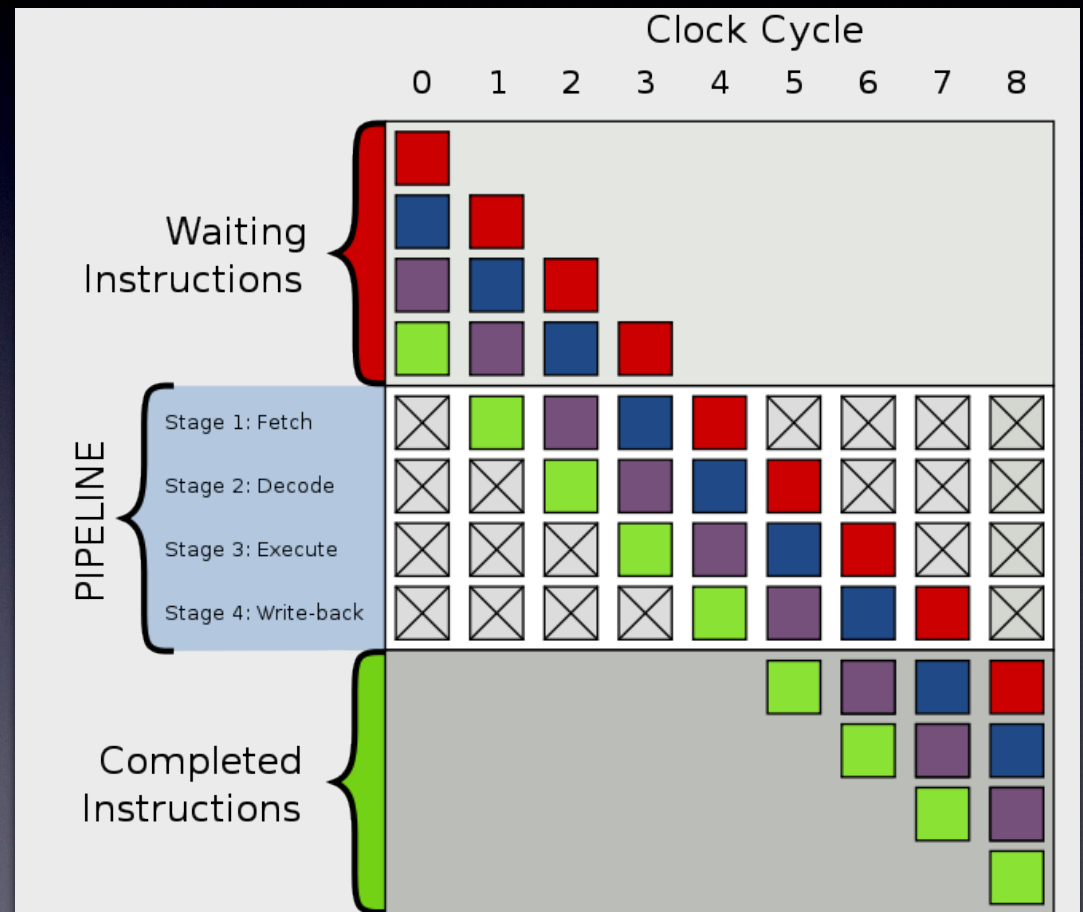


# Branch Predictors

# First some motivation

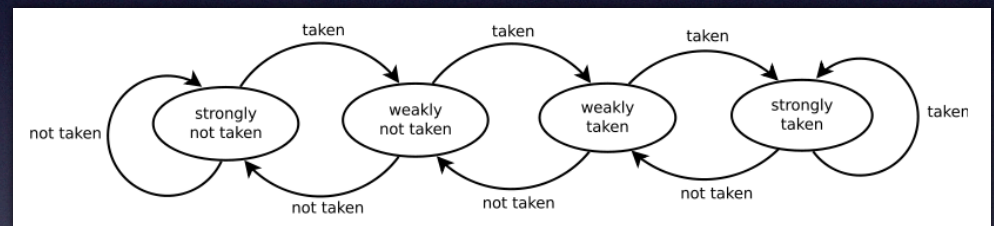
- Branch prediction is necessary because of Instruction Pipelining
- Without branch prediction the pipeline would stall until the condition was decided
- The cost of a misprediction is proportional to the length of the pipeline





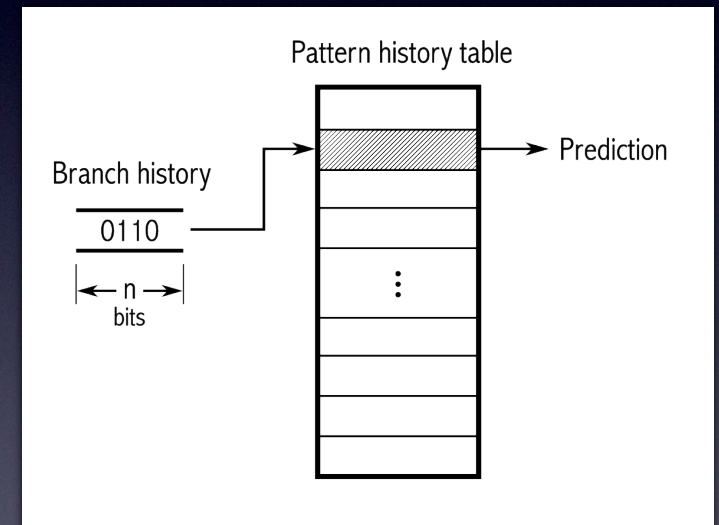
# Static and 'last time' strategies

- Static prediction
- Last time predictor
- Saturating counter



# Correlating Strategies

- Two-level adaptive predictor
- Local
  - Per branch history register
  - PC indexes the correct history register





# Correlating Strategies

- Global

```
if (cond1)
```

```
...
```

```
if (cond1 && cond2)
```

first not taken =>

```
second not taken
```

```
-----  
if (aa==2) aa = 0;
```

```
// B1
```

```
if (bb==2) bb = 0;
```

```
// B2
```

```
if (aa!=bb) { ... }
```

```
// B3
```

```
// if B1 and B2 not taken (i.e. both fall  
through) => clearly B3 must be taken (i.e. will  
jump past the block
```

# Correlating Strategies

- Hybrid
  - Combination of different strategies. (use the best)
  - Meta predictor to decide which to use.