<u>Input:</u> scalar $d = (d_{n-1} d_{n-2} \dots d_1 d_0)_2$, point P

<u>Output:</u> $d \cdot P$

1. $R_0 := \Theta$, $R_1 := P$
2. for $j := n-1$ downto 0 do
3. $\quad R_{1-d_j} := R_{1-d_j} + R_{d_j}$
4. $\quad R_{d_j} := 2 \cdot R_{d_j}$
5. Return R_0.

Note that in each iteration both point addition and point doubling are performed – side channel attacks are expected to be more difficult.

<u>Example:</u> let $d = 22 = (\overset{4}{1}\,\overset{3}{0}\,\overset{2}{1}\,\overset{1}{1}\,\overset{0}{0})_2$

Initially: $R_0 := \Theta$, $R_1 := P$

$j=4$: $R_0 := P$, $R_1 := 2P$

$j=3$: $R_1 := 3P$, $R_0 := 2P$

$j=2$: $R_0 := 5P$, $R_1 := 6P$

$j=1$: $R_0 := 11P$, $R_1 := 12P$

$j=0$: $R_1 := 23P$, $\underline{R_0 := 22P}$

Indeed, the result is in R_0.
Interestingly, $R_1 = R_0 + P$, and this equality is invariant of the computations above.
The optimizations of the EC arithmetic take advantage of the following observations:

- Each iteration of the Montgomery Ladder requires point addition and point doubling - hence they can be implemented in a single procedure ECADDDBL.
- It is possible to remove y-coordinate from point doubling formula, and if the difference $P_3' = P_1 - P_2$ is known then y-coordinate can also be neglected in the formula for $P_3 = P_1 + P_2$.

Note that in the Montgomery Ladder $R_1 - R_0 = P$ in every iteration of the algorithm.

To emphasise that ECADDDBL does not make use of y-coordinate we denote the procedure by xECADDDBL.

<u>Input</u>: • $d = (d_{n-1}\, d_{n-2} \ldots d_1, d_0)_2$, where $d_{n-1} = 1$,

• point P

<u>Output</u>: $d \cdot P$

point doubling procedure, does not need y-coordinate

1. $Q[0] := P$, $Q[1] = xECDBL(P)$
2. for $i = n-2$ downto 0
3. $(Q[d_i \text{ xor } 1], \underbrace{Q[d_i]}_{\text{the result of point doubling}}) :=$

   $:= xECADDDBL(\underbrace{Q[d_i]}_{\text{this element will be doubled}}, Q[d_i \text{ xor } 1])$
4. Return $Q[0]$.

The algorithm above will utilize the Standard Projective Coordinates. But instead of $P=(x:y:1)$ in the first line we take its randomized representation without y-coordinate, that is $P=(rx:r)$ for $r \in K^*$.

The difference $P_3' = Q[1]-Q[0]$ will be represented as $P_3'=(x:1)$. That is, $X_3'=x$, $Z_3'=1$.

Let us derive the formulas for point addition and point doubling without y-coordinate:

$$P_3=(x_3, y_3)=P_1+P_2$$

$$P_3'=(x_3', y_3')=P_1-P_2,$$

where $P_1=(x_1, y_1)$ $P_2=(x_2, y_2)$

go to my handwritten notes ;)

Let P be the input to the last version of the Montgomery ladder algorithm (the version that utilizes xECDBL, xECADDDBL procedures).

Assume that $P=(x,y)$ does not satisfy the elliptic curve E equation:

$$(1) \quad y^2=x^3+ax+b$$

but the equation

$$(10) \quad d y^2=x^3+ax+b$$

of the twist E^d, for some fixed d being QN in $\mathbb{F}_p^*$ (quadratic non-residue in $\mathbb{F}_p^*$)

It is easy to find such point P:

Define $f_{a,b}(x) = x^3 + ax + b$

1) if for a random $x \in \mathbb{F}_p$ $\omega = f_{a,b}(x)$ is such that $\omega = 0$ or ω is QR in $\mathbb{F}_p^*$, then we set

$$y = \sqrt{\omega} \quad \text{or} \quad y = -\sqrt{\omega} = $$
$$= p - \sqrt{\omega}$$

so $y \in \mathbb{F}_p$ and $(x, y) \in E$.

2) If $\omega = f_{a,b}(x)$ is QN in $\mathbb{F}_p^*$ then it means that ω is an odd power of a generator $g \in \mathbb{F}_p^*$.

That is $\omega = g^l$, l is odd.

The same applies to d defining E^d (for any fixed d such that d is QN in $\mathbb{F}_p^*$), that is

$$d = g^k, \quad k \text{ is odd}$$

Then $\omega \cdot d^{-1} = g^{l-k}$ is an even power of the generator g

$$= g^{\underbrace{(l-k) + \cancel{t} \cdot (p-1)}_{\text{to get positive integer}}}$$

$t \in \{0, 1\}$

so it is a QR in $\mathbb{F}_p^*$.

We set $y = \sqrt{\omega \cdot d^{-1}}$ or $y = p - \sqrt{\omega d^{-1}}$

See that $y \in \mathbb{F}_p^*$ and (x, y) satisfies (10), the equation for E^d.

What is the probability of picking a random x from $\mathbb{F}_p^*$ such that there exist $y \in \mathbb{F}_p^*$ such that $(x, y) \in E^d$?

For a random x we estimate the probability that $w = f_{a,b}(x) \in \mathbb{F}_p^*$ is a QN by:

$$\approx \frac{\frac{p-1}{2}}{p-1} = \frac{1}{2}$$

That is, for large p and a random $x \in \mathbb{F}_p$ we get, with probability $\approx \frac{1}{2}$ a point $P = (x, y) \in E^d$ such that $y \in \mathbb{F}_p^*$.

If such Point P is an input to the Montgomery Ladder algorithm then the implementation does not notice this:

- there is no check that the input point belongs to E,
- the procedures xECDBL, xECADDDBL do not make use of y-coordinate
- the YRecovering procedure will yield us $(X : Y : Z)$ such that $(x, y) = (X/Z, Y/Z)$ is on E^d, but there is no check that the output belongs to E.

P-224 - the twist of this curve has smooth order!